

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Комп'ютеризована система контролю клімату та вологості ґрунту в домашній теплиці

Виконав: студент IV курсу, групи СНС-42

спеціальності 122 Комп'ютерні науки

(шифр і назва спеціальності)

Мезей Д.І.

(підпис)

(прізвище та ініціали)

Керівник

(підпис)

Марценко С.В.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Марценко С.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

Загородна Н.В.

(прізвище та ініціали)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.
(підпис) (прізвище та ініціали)

« » 2024 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Мезей Дмитро Іванович
(прізвище, ім'я, по батькові)

1. Тема роботи Комп'ютеризована система контролю клімату та вологості ґрунту в домашній теплиці

Керівник роботи Марценко Сергій Володимирович, к. т. н., доцент кафедри КН
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «29» квітня 2024 року № 4/7-472

2. Термін подання студентом завершеної роботи 24 червня 2024р.

3. Вихідні дані до роботи Літературні та інтернет джерела інформації про системи контролю клімату та вологості ґрунту. Технічна документація по обладнанню ESP32 C3 та програмному забезпеченню.

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1. Огляд аналогів системи контролю клімату та вологості ґрунту в домашній теплиці
Аналіз промислових та аматорських рішень. Переваги та недоліки різних технологій обміну даними . Висновки до першого розділу. 2. Проектування системи контролю клімату та вологості ґрунту в домашній теплиці. Вибір основних компонентів: контролера, модулів, давачів. Розробка електричної принципової схеми підключення. Висновки до другого розділу. 3. Налаштування та тестування системи. Написання програмного коду. Монтаж та перевірка системи. Висновки до третього розділу. 4. Безпека життєдіяльності, основи охорони праці. Висновки. Перелік джерел. Додатки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Титульна сторінка. 2. Актуальність дослідження. 3. Мета, Об'єкт, Предмет дослідження.
4. Завдання дослідження. 5. Схеми взаємодії елементів системи. 6. Компоненти системи.
7. Компоненти системи. 8. Компоненти системи. 9. Вибір мови програмування для ESP32 C3.
10. Програмування ESP32 C3 у Arduino IDE. 11. Тестування системи на базі ESP32 C3.
12. Змонтована система. 13. Робота системи поливу. 14. Активація зволоження повітря.
15. Висновки. 16. Завершальний.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	Сенчишин В. С., к.т.н. доцент кафедри МТ	12.06.2024	14.06.2024

7. Дата видачі завдання 29 січня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	30.01.2024	Виконано
2.	Підбір джерел по темі кваліфікаційної роботи	31.01.2024-03.02.2024	Виконано
3.	Опрацювання джерел по темі кваліфікаційної роботи	04.02.2024-06.02.2024	Виконано
4.	Виконання дослідження щодо розробки комп'ютеризованої системи контролю клімату та вологості ґрунту в домашній теплиці.	07.02.2024-11.02.2024	Виконано
5.	Оформлення розділу «Огляд предметної області»	03.06.2024-05.06.2024	Виконано
6.	Оформлення розділу «Елементи системи контролю клімату»	06.06.2024-08.06.2024	Виконано
7.	Оформлення розділу «Налаштування та тестування системи»	09.06.2024-11.06.2024	Виконано
8.	Виконання завдання до підрозділу «Безпека життєдіяльності»	12.06.2024-13.06.2024	Виконано
9.	Виконання завдання до підрозділу «Основи охорони праці»	14.06.2024-15.06.2024	Виконано
10.	Оформлення кваліфікаційної роботи	16.06.2024-17.06.2024	Виконано
12.	Нормоконтроль	18.06.2024-19.06.2024	Виконано
13.	Перевірка на плагіат	20.06.2024	Виконано
14.	Попередній захист кваліфікаційної роботи	21.06.2024	Виконано
15.	Захист кваліфікаційної роботи	30.06.2024	

Студент

_____ (підпис)

Мезей Д.І.

_____ (прізвище та ініціали)

Керівник роботи

_____ (підпис)

Марценко С.В.

_____ (прізвище та ініціали)

АНОТАЦІЯ

Комп'ютеризована система контролю клімату та вологості ґрунту в домашній теплиці // Кваліфікаційна робота освітнього рівня «Бакалавр» // Мезей Дмитро Іванович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНс-42 // Тернопіль, 2024 // С. 69, рис. – 16, табл. – 0, кресл. – 16, додат. – 1, бібліогр. – 35.

Ключові слова: Веб інтерфейс, автоматизація, теплиця, ESP32 С3, системи контролю, датчики.

Кваліфікаційна робота присвячена розробці і валідації системи контролю клімату та вологості ґрунту для домашньої теплиці з використанням мікроконтролера ESP32 С3.

У першому розділі проведено аналіз сучасних промислових та аматорських систем моніторингу. Показано можливості комунікаційних технологій, таких як MQTT і WebSocket, для інтеграції різноманітних системних компонентів.

У другому розділі описано детальне проектування системи контролю, включаючи вибір і інтеграцію датчиків, таких як DHT11, FC-28 та HW-038 для вимірювання температури, вологості повітря і ґрунту та рівня води. Також розглянуто використання мікроконтролера, як центрального елемента системи.

Третій розділ зосереджено на програмній реалізації та тестуванні системи, де аналізується ефективність програмного коду, розробленого у середовищі Arduino IDE для взаємодії з апаратними компонентами і користувацьким веб інтерфейсом.

ANNOTATION

A computerized system for controlling climate and soil moisture in a home greenhouse // Qualification work of the educational level "Bachelor" // Mezei Dnytro // Ternopil Ivan Pulyu National Technical University, Computer and Information Systems and Software Engineering Faculty, Computer Sciences Department, group SNs-42 // Ternopil, 2024 // P. 69, fig. - 16, tabl. - 0, chair. - 16, annexes. – 1, references - 35.

Keywords: Web interface, automation, greenhouse, ESP32 C3, control systems, sensors.

The qualification work is devoted to the development and validation of a climate and soil moisture control system for a home greenhouse using the ESP32 C3 microcontroller.

The first chapter analyzes modern industrial and amateur monitoring systems. The possibilities of communication technologies, such as MQTT and WebSocket, for integrating various system components are shown.

The second section describes the detailed design of the monitoring system, including the selection and integration of sensors such as DHT11, FC-28, and HW-038 for measuring temperature, air and soil humidity, and water level. The use of a microcontroller as the centerpiece of the system is also discussed.

The third section focuses on the software implementation and testing of the system, where the effectiveness of the program code developed in the Arduino IDE environment for interacting with the hardware components and the user web interface is analyzed.

ПЕРЕЛІК СКОРОЧЕНЬ

BLE (англ. Bluetooth Low Energy) – Bluetooth з низьким енергоспоживанням.

GPIO (англ. General-purpose input/output) – Інтерфейс введення/виведення загального призначення.

IoT (англ. Internet of Things) – Інтернет речей.

MQTT (англ. Message Queuing Telemetry Transport) – Телеметрична пересилка повідомлень з чергою.

Wi-Fi (англ. Wireless Fidelity) – Бездротовий стандарт передачі даних.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ	10
1.1 Огляд промислових рішень систем контролю клімату	10
1.2 Огляд аматорських рішень систем контролю клімату	14
1.3 Порівняння технологій обміну даними для пристроїв IoT	17
1.4 Висновок до першого розділу	20
РОЗДІЛ 2. ЕЛЕМЕНТИ СИСТЕМИ КОНТРОЛЮ КЛІМАТУ	22
2.1 Схема взаємодії елементів системи.....	22
2.2 Опис та характеристика елементів автоматизованої теплиці.....	23
2.2.1. Мікроконтролер ESP32-C3.....	23
2.2.2. Давач температури та вологості DHT11	26
2.2.3. Гігрометр FC-28 та компаратор напруги LM393	27
2.2.4. Давач рівня води HW-038	29
2.2.5. Водяний насос RS-360SH.....	31
2.2.6. Транзисторний модуль MOSFET IRF520N	32
2.2.7. Водонепроникний корпус SONOFF	33
2.2.8. Ультразвуковий зволожувач повітря	34
2.3 Електрична принципова схема системи.....	35
2.4 Висновок до другого розділу	37
РОЗДІЛ 3. НАЛАШТУВАННЯ ТА ТЕСТУВАННЯ СИСТЕМИ	38
3.1 Вибір мови програмування для ESP32 C3	38
3.2 Опис програмного коду для ESP32 C3	38
3.3 Монтаж та перевірка працездатності системи	51
3.4 Висновок до третього розділу	57
РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	58
4.1 Природне середовище і його забруднення	58
4.2 Рекомендації щодо естетичного оформлення інтерєру цеху, дільниці.....	60

4.3 Висновок до четвертого розділу	62
ВИСНОВКИ.....	64
ПЕРЕЛІК ДЖЕРЕЛ.....	66
ДОДАТКИ	

ВСТУП

Актуальність теми. Ефективність використання ресурсів і стійкість екосистем є ключовими пріоритетами в умовах зростаючого впливу людської діяльності на довкілля. Важливість розробки автоматизованих систем моніторингу для домашнього господарства зростає. Інтеграція технологій IoT відкриває нові перспективи для підвищення ефективності управління ресурсами, особливо у тепличному господарстві, забезпечуючи більш стабільне та продуктивне вирощування культур.

Мета і задачі дослідження. Метою даного дослідження є розробка та валідація інтегрованої системи контролю клімату та вологості ґрунту для домашньої теплиці, використовуючи мікроконтролер ESP32 C3.

Задачі дослідження включають:

- Аналіз сучасних промислових та аматорських систем управління.
- Проектування системи контролю клімату та вологості ґрунту.
- Інтеграція мікроконтролера та датчиків для контролю кліматичних параметрів.
- Розробка програмного забезпечення для збору та аналізу даних.
- Тестування та оптимізація системи для забезпечення точності контролю.

Об'єкт дослідження. Об'єктом дослідження є процес моніторингу та управління кліматичними умовами в домашній теплиці, який виконується за допомогою спеціалізованого мікроконтролерного обладнання.

Предмет дослідження. Предметом дослідження є інтегровані технології та методи автоматизації, використані у системі моніторингу і контролю, зокрема використання мікроконтролера ESP32 C3, датчиків, а також комунікаційний протокол такий, як WebSocket.

Практичне значення одержаних результатів. Результати дослідження мають практичний значення для контролю кліматичних умов у домашніх теплицях, що сприяє оптимізації вирощування культур та зниженню витрат на

воду і енергію. Розроблена система може служити як основа для подальших інновацій у галузі автоматизації невеликих тепличних господарств, що відкриває можливості для більш екологічного та економічно ефективного сільського господарства.

РОЗДІЛ 1. ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд промислових рішень систем контролю клімату

Технології відіграють важливу роль у багатьох аспектах нашого життя, включаючи сільське господарство. Зокрема, використання просунутих систем автоматизації в теплицях відкриває нові можливості для оптимізації вирощування рослин, зниження витрат та підвищення стабільності врожайності. Наступна стаття розглядає три популярні системи автоматизації теплиць: ControlByWeb X-600M, Sensaphone Sentinel та Argus LIVE [1]. Кожна з цих систем пропонує унікальний набір функціональних можливостей, які можуть значно покращити процес вирощування рослин у контрольованому середовищі.

Промислові системи автоматизації теплиць стали незамінними для сучасного сільського господарства, оскільки вони дозволяють оптимізувати процеси вирощування рослин, зменшити витрати на робочу силу та забезпечити стабільні врожаї.

ControlByWeb X-600M є багатофункціональною модульною системою, яка дозволяє контролювати і моніторити різноманітні параметри теплиці, включаючи температуру, вологість, освітлення, полив та вентиляцію [1]. Ця система має можливість підключення до мережі Інтернет, що дозволяє користувачам віддалено контролювати параметри теплиці за допомогою веб-інтерфейсу.

Однією з ключових переваг ControlByWeb X-600M є його модульність. Користувачі можуть налаштовувати систему відповідно до своїх конкретних потреб, додаючи або видаляючи модулі залежно від вимог. Наприклад, можна підключити модулі для моніторингу температури та вологості, модулі для контролю освітлення та поливу, а також модулі для управління вентиляцією. Це

дозволяє створювати індивідуальні рішення для кожної теплиці, забезпечуючи оптимальні умови для вирощування рослин.

Система підтримує різноманітні датчики та виконавчі пристрої, що робить її надзвичайно гнучкою. Датчики можуть вимірювати температуру, вологість, рівень CO₂, інтенсивність освітлення та інші параметри, важливі для зростання рослин. Виконавчі пристрої, такі як насоси, вентилятори та світильники, можуть бути автоматично керовані системою на основі зібраних даних. Це дозволяє забезпечити точний контроль за умовами в теплиці та оперативно реагувати на будь-які зміни [2].

ControlByWeb X-600M також має потужні функції моніторингу та звітності. Система може зберігати історію даних про стан теплиці, що дозволяє користувачам аналізувати ці дані та приймати обґрунтовані рішення для покращення умов вирощування. Крім того, система підтримує відправку сповіщень про тривоги через електронну пошту або SMS, що забезпечує оперативне інформування про будь-які аномалії.

Ще однією важливою особливістю ControlByWeb X-600M є можливість інтеграції з іншими системами та пристроями. Це дозволяє створювати комплексні рішення для автоматизації теплиць, об'єднуючи різноманітні системи управління в єдину мережу. Наприклад, можна інтегрувати систему з існуючими системами опалення або охолодження, забезпечуючи повний контроль за мікрокліматом у теплиці.

Sensaphone Sentinel передова система автоматизації та моніторингу теплиць, що забезпечує надійний контроль за ключовими параметрами середовища [3]. Ця система відома своєю здатністю відправляти попередження в реальному часі, що дозволяє користувачам оперативно реагувати на можливі проблеми та підтримувати оптимальні умови для вирощування рослин.

Sensaphone Sentinel моніторить такі параметри, як температура, вологість, рівень CO₂, освітлення та інші важливі показники. Система оснащена різноманітними датчиками, які можуть бути налаштовані відповідно до конкретних потреб користувача. Всі зібрані дані зберігаються на хмарній

платформі, що забезпечує доступ до інформації з будь-якої точки світу за допомогою мережі Інтернет [4]. Це дозволяє користувачам контролювати стан теплиці.

Однією з ключових особливостей Sensaphone Sentinel є її здатність надсилати повідомлення про тривогу у разі виявлення аномалій. Користувачі можуть отримувати сповіщення через електронну пошту, SMS або телефонний дзвінок. Це забезпечує швидку реакцію на можливі проблеми, що можуть виникнути в теплиці, наприклад, у разі різкого підвищення або зниження температури, що може негативно вплинути на рослини. Така функціональність є особливо корисною для великих тепличних господарств, де оперативне реагування на зміни умов є критичним для підтримки стабільної врожайності.

Система Sensaphone Sentinel відрізняється своєю простотою у встановленні та використанні. Вона не потребує складного налаштування або спеціальних технічних знань, що робить її доступною для широкого кола користувачів. Інтуїтивно зрозумілий інтерфейс дозволяє легко налаштовувати параметри моніторингу та отримувати доступ до зібраних даних. Крім того, система підтримує інтеграцію з іншими пристроями та системами, що дозволяє створювати комплексні рішення для автоматизації теплиць.

Sensaphone Sentinel також забезпечує високий рівень безпеки даних. Всі зібрані дані передаються на хмарну платформу зашифрованими каналами, що забезпечує захист від несанкціонованого доступу [5]. Крім того, система зберігає історичні дані, що дозволяє користувачам аналізувати інформацію за певний період та приймати обґрунтовані рішення для покращення умов вирощування.

Argus LIVE передова система автоматизації теплиць, що забезпечує комплексний підхід до моніторингу та управління всіма аспектами тепличного господарства [6]. Ця система розроблена для того, щоб оптимізувати процеси вирощування рослин, підвищити врожайність та забезпечити стабільні результати, використовуючи сучасні технології та інноваційні рішення.

Однією з ключових особливостей Argus LIVE є її здатність контролювати широкий спектр параметрів, включаючи температуру, вологість, освітлення, полив, вентиляцію та рівень CO₂. Система використовує передові датчики для збору точних даних у режимі реального часу, що дозволяє оперативно реагувати на будь-які зміни умов в теплиці. Завдяки цьому можна забезпечити оптимальні умови для зростання рослин та зменшити ризики, пов'язані з екстремальними змінами мікроклімату [7].

Argus LIVE використовує інтелектуальні алгоритми для автоматичного управління різними системами теплиці. Наприклад, система може автоматично регулювати освітлення залежно від інтенсивності природного світла або контролювати полив на основі рівня вологості ґрунту. Це дозволяє значно зменшити витрати на ресурси та підвищити ефективність використання води та енергії. Крім того, автоматизація рутинних процесів знижує потребу в ручній праці, що є важливим фактором для великих тепличних господарств.

Однією з найважливіших переваг Argus LIVE є можливість інтеграції з іншими системами управління. Це дозволяє створювати комплексні рішення для автоматизації, об'єднуючи різні аспекти управління в єдину систему. Наприклад, можна інтегрувати Argus LIVE з системами опалення та охолодження, забезпечуючи повний контроль за мікрокліматом у теплиці. Така інтеграція дозволяє досягти максимальної ефективності та стабільності в умовах вирощування рослин.

Argus LIVE також надає потужні інструменти для аналізу даних та звітності. Користувачі можуть отримувати детальні звіти про стан теплиці, аналізувати історичні дані та виявляти тенденції, що можуть впливати на врожайність. Це дозволяє приймати обґрунтовані рішення на основі реальних даних та оптимізувати процеси вирощування. Крім того, система може надсилати сповіщення про аномалії через електронну пошту або SMS, що забезпечує швидку реакцію на можливі проблеми.

Argus LIVE відрізняється високим рівнем надійності та безпеки. Всі дані передаються зашифрованими каналами, що забезпечує захист від

несанкціонованого доступу. Система також має можливість резервного копіювання даних, що гарантує їх збереження у разі збоїв або технічних проблем.

1.2 Огляд аматорських рішень систем контролю клімату

Сучасні технології змінюють підходи до ведення сільського господарства, дозволяючи фермерам забезпечувати сталість та ефективність вирощування культур. Особливо помітний вплив автоматизації на галузь тепличного господарства, де краплинне зрошення стає ключовим елементом для оптимізації ресурсів [8]. Наступний огляд присвячено системі керування та моніторингу для автоматизації краплинного зрошення у теплицях, яка використовує сучасні комп'ютерні технології та інтеграцію різноманітних датчиків для точного контролю над вологістю, температурою та іригацією.

Система, розроблена Elaydi, є прикладом інноваційного підходу до управління теплицями, що відкриває нові можливості для підвищення продуктивності та зниження екологічного впливу вирощування рослин.

У статті [9] розглядається комплексна розробка та випробування системи керування та моніторингу, призначеної для автоматизації процесу краплинного зрошення у теплицях. Система використовує комп'ютерне управління та карту збору даних, з'єднуючи кілька датчиків для забезпечення точного контролю над вологістю, температурою та іригацією.

Теплиці традиційно використовуються для оптимізації вирощування культур, дозволяючи фермерам культивувати рослини незалежно від сезонних обмежень. Однак управління ресурсами, такими як вода і світло, може бути складним без використання сучасних технологій. Система, представлена Elaydi, автоматизує краплинне зрошення, використовуючи датчики вологості та температури для регулювання поливу відповідно до потреб рослин, що значно підвищує ефективність використання води та сприяє здоровому росту культур.

Система моніторингу та контролю базується на програмному забезпеченні LABVIEW, яке використовується для візуалізації даних та управління процесами в реальному часі. Це дозволяє фермерам не тільки спостерігати за станом теплиці, але й дистанційно керувати налаштуваннями через інтернет, що є величезною перевагою для господарств, розташованих у віддалених або труднодоступних місцях.

Серед ключових елементів системи – датчики вологості та температури, які передають дані на головний комп'ютер через карту збору даних. Ця інформація використовується для автоматичного регулювання системи поливу, включаючи клапани та насоси, що оптимізують водоспоживання та забезпечують достатній рівень вологості для культур. Контрольні панелі налаштовуються таким чином, щоб користувачі могли легко змінювати параметри залежно від типу вирощуваних культур, а також підлаштовуватися під різні погодні умови.

Іншою особливістю системи є її здатність до самостійної роботи без постійного втручання користувача. Такий підхід ідеально підходить для новачків у сфері фермерства, які можуть не мати глибоких технічних знань. Автоматичні сповіщення та електронні листи про критичні зміни умов в теплиці дозволяють оперативно реагувати на можливі проблеми, забезпечуючи додатковий рівень безпеки та контролю.

Система також включає можливості для збору та аналізу даних, що можуть використовуватися для вивчення довгострокових трендів у вирощуванні та для оптимізації виробничих процесів. Збір даних допомагає виявляти неефективне використання ресурсів і сприяє впровадженню більш сталого підходу до управління водними ресурсами.

Автор у роботі [10] використовує передові мікроконтролери та вбудовані системи для моніторингу та контролю різних параметрів в теплиці, таких як температура, вологість, рівень CO₂ і освітленість [11]. Основна ідея полягає в тому, щоб зробити цю систему доступною, ефективною та економічною, що особливо важливо для фермерів у регіонах з обмеженими ресурсами.

Однією з ключових переваг системи є її гнучкість. Використання платформ DIY дозволяє фермерам самостійно налаштовувати систему під конкретні умови своїх теплиць. Крім того, система включає використання відновлюваних джерел енергії, таких як фотоелектричні батареї, що забезпечують живлення не тільки для вбудованих систем, але й для мобільних інтернет-модемів, що є істотним на сільських територіях.

Система також підтримує зв'язок через мобільний інтернет, дозволяючи фермерам контролювати теплицю дистанційно з будь-якої точки світу через веб-інтерфейс або смартфон. Це важливо для оперативного реагування на зміни умов всередині теплиці та вжиття необхідних заходів без затримок [12].

Автори використали декілька протоколів для передачі даних, таких як RS-485, CAN, Bluetooth, GPRS, GSM, кожен з яких вибирається залежно від вимог до відстані та швидкості передачі даних [13]. Це дозволяє оптимізувати вартість установки системи та забезпечити її ефективність.

Система використовує передові технології IoT, що дозволяє інтегрувати різні вбудовані системи та датчики в єдину мережу. Такий підхід забезпечує можливість збору та аналізу великої кількості даних, що є критично важливим для оптимізації умов у теплицях і підвищення врожайності.

Дизайн системи також включає розробку бази даних та інтерфейсів для користувачів, що дозволяє зберігати, аналізувати та візуалізувати історичні дані. Це не тільки спрощує моніторинг та управління процесами у теплиці, але й дозволяє проводити аналіз для планування майбутніх дій.

Система, описана у джерелі [14] базується на Arduino Uno, платі на базі мікроконтролера ATmega328, яка обладнана численними входами та виходами для підключення різних датчиків, таких як датчик температури, вологості, освітленості та вмісту CO₂ у повітрі [15]. Ці параметри є вирішальними для забезпечення оптимальних умов для росту рослин у теплицях, де важливо підтримувати строгий контроль над мікрокліматом.

Використання Arduino дозволяє збирати дані з датчиків, обробляти їх та, на основі отриманих даних, автоматично регулювати умови в теплиці, зокрема,

активувати системи поливу або освітлення. Система передбачає використання графічного інтерфейсу користувача через LabVIEW, що спрощує взаємодію з системою для користувачів без технічної освіти.

Значущість розробки полягає у здатності системи надавати детальну інформацію про стан теплиці в реальному часі, що дозволяє фермерам приймати своєчасні рішення для оптимізації умов росту рослин. Це особливо важливо у сучасних умовах урбанізації та зменшення кількості доступної землі для сільськогосподарських потреб.

Завдяки автоматизації, система мінімізує людські помилки та знижує витрати на працю, необхідні для підтримки оптимальних умов у теплиці. Це робить технологію особливо корисною для малих фермерських господарств, де ресурси обмежені, а потреба в оптимізації використання води та енергії є критичною.

Інноваційність даної розробки полягає також у гнучкості системи, що може бути адаптована під різні умови та різні типи рослин, що вирощуються у теплицях. Використання стандартних компонентів, таких як Arduino та прості давачі, робить систему відносно дешевою та легкою для масштабування та адаптації під конкретні потреби користувача.

1.3 Порівняння технологій обміну даними для пристроїв ІоТ

В епоху розумних технологій, вибір правильних комунікаційних протоколів є ключовим аспектом успішної реалізації систем домашньої автоматизації. Ці системи вимагають стабільних, безпечних та ефективних методів зв'язку для взаємодії між безліччю пристроїв у розумному будинку. Огляд трьох популярних технологій обміну даними - MQTT, WebSocket і ZigBee - допоможе зрозуміти, як кожен з них може служити різним потребам у контексті домашньої автоматизації, забезпечуючи надійність, безпеку, і зручність управління. [16].

У контексті домашньої автоматизації важливим є вибір правильних протоколів зв'язку, щоб забезпечити надійність, безпеку та зручність використання системи. Розглянемо три популярні технології: MQTT, WebSocket і ZigBee, які часто використовуються у цій сфері.

MQTT є легким протоколом обміну повідомленнями, що підтримує структуру "публікація-підписка" [17]. Він був спеціально розроблений для сценаріїв з обмеженими ресурсами та низькою пропускнуою здатністю, що часто зустрічається у пристроях IoT. Протокол дозволяє пристроям відправляти повідомлення до центрального брокера, який, в свою чергу, пересилає ці повідомлення підписаним клієнтам. Це знижує витрати на зв'язок і спрощує архітектуру системи, оскільки пристрої не мають встановлювати між собою безпосередні з'єднання. MQTT чудово підходить для застосувань, де потрібна висока надійність та ефективність доставки повідомлень при мінімальному обміні даними, наприклад, у системах дистанційного керування. [18].

WebSocket, у свою чергу, надає можливість двостороннього зв'язку в реальному часі між клієнтом та сервером через веб-браузери. Це відкриває двері для більш динамічних та інтерактивних веб-додатків, де користувачі можуть отримувати постійні оновлення без необхідності відновлювати сторінку [19]. WebSocket особливо ефективний для таких сценаріїв, як онлайн-ігри, торгівля в реальному часі, або коли потрібне швидке спілкування, як наприклад, у чатах. Основна перевага WebSocket полягає у тому, що він дозволяє веб-додаткам відправляти та отримувати дані з мінімальними затримками, оскільки не потребує повторного встановлення з'єднань, як у випадку з традиційним HTTP-запитом [20].

У порівнянні, MQTT і WebSocket служать різним потребам і вирішують різні проблеми в комунікаційних системах. MQTT оптимізований для мінімальної витрати зв'язку та низького енергоспоживання, що робить його ідеальним для пристроїв IoT з обмеженою енергією та мережевими ресурсами. WebSocket призначений для сценаріїв, що вимагають високої інтерактивності

та мінімальної відповіді сервера, що робить його ідеальним для більш комплексних веб-додатків, які взаємодіють з користувачем в реальному часі.

ZigBee є бездротовим протоколом зв'язку, який вирізняється своєю енергоефективністю та здатністю до створення великих мережевих топологій, що забезпечують надійне покриття в складних умовах [21]. Розроблений на основі стандарту IEEE 802.15.4, ZigBee часто використовується в промислових автоматизаційних системах, домашній автоматизації, розумних будинках та медичних моніторингових системах. Його головні переваги включають низьке споживання енергії, що дозволяє пристроям працювати на батарейках роками без заміни, а також можливість створення масштабованих мереж з сотень і тисяч вузлів.

ZigBee підтримує декілька типів мережевих топологій, зокрема зіркову, деревоподібну та меш (сітчасту). Саме меш-топология дозволяє кожному вузлу комунікувати з множиною інших вузлів безпосередньо або через проміжні вузли, які служать ретрансляторами, значно збільшуючи дальність та надійність мережі. Така архітектура особливо корисна в будівлях зі складними просторовими умовами, де безпосередні сигнали можуть бути заблоковані або ослаблені стінами та іншими перешкодами.

Протокол ZigBee забезпечує високий рівень безпеки завдяки використанню шифрування AES-128. Це особливо важливо для застосувань, де вимагається захист чутливих даних, як наприклад, в системах керування доступом або при передачі особистих медичних даних. Також протокол включає механізми аутентифікації та цілісності даних, що запобігають несанкціонованому доступу та забезпечують достовірність переданої інформації.

Однією з ключових особливостей ZigBee є його здатність до самоорганізації та самовідновлення мережі. Вузли можуть автоматично підключатися до мережі та вибирати оптимальний шлях для передачі даних, що спрощує налаштування та підтримку системи. У випадку відмови одного з

вузлів, мережа може автоматично переконфігурувати себе для підтримки безперервної роботи.

Вибір між MQTT, WebSocket та ZigBee залежить від конкретних потреб домашньої автоматизації. MQTT є оптимальним вибором для простих систем, де важливі надійність та мінімальна затримка. WebSocket підходить для додатків, які потребують високої взаємодії з користувачем через веб-інтерфейси. ZigBee ідеально підходить для комплексних систем домашньої автоматизації з великою кількістю пристроїв, які потребують низького споживання енергії та високого рівня безпеки.

1.4 Висновок до першого розділу

В першому розділі кваліфікаційної роботи присвячений аналізу сучасних систем моніторингу та управління у сфері тепличного господарства та домашньої автоматизації. Зокрема, було розглянуто як промислові системи, так і аматорські рішення, що дозволяє зрозуміти широкий спектр доступних технологій та їх застосування.

Промислові системи, такі як ControlByWeb X-600M, Sensaphone Sentinel та Argus LIVE, забезпечують високий рівень контролю та інтеграції, що є критично важливим для великих тепличних комплексів. Ці системи забезпечують не тільки автоматизацію основних процесів, але й моніторинг важливих параметрів з можливістю швидкого реагування на зміни умов. Їх застосування може значно підвищити ефективність використання ресурсів і стабільність вирощування.

На відміну від промислових рішень, аматорські системи на базі Arduino та інших мікроконтролерів пропонують більш доступні, але водночас ефективні варіанти для індивідуальних потреб. Вони забезпечують гнучкість у налаштуванні і можливість адаптації до специфічних умов, що є привабливим для невеликих ферм або особистих господарств.

Ключовим аспектом ефективності обох типів систем є вибір відповідних технологій передачі даних. MQTT, WebSocket, ZigBee та інші технології зв'язку відіграють важливу роль у забезпеченні надійності та безпеки передачі даних, а також у взаємодії різних компонентів систем. Вибір технології залежить від конкретних вимог до швидкості, надійності, відстані передачі даних та енергоспоживання.

РОЗДІЛ 2. ЕЛЕМЕНТИ СИСТЕМИ КОНТРОЛЮ КЛІМАТУ

2.1 Схема взаємодії елементів системи

На рисунку 2.1 показано схему взаємодії елементів система контролю клімату та вологості ґрунту в домашній теплиці.

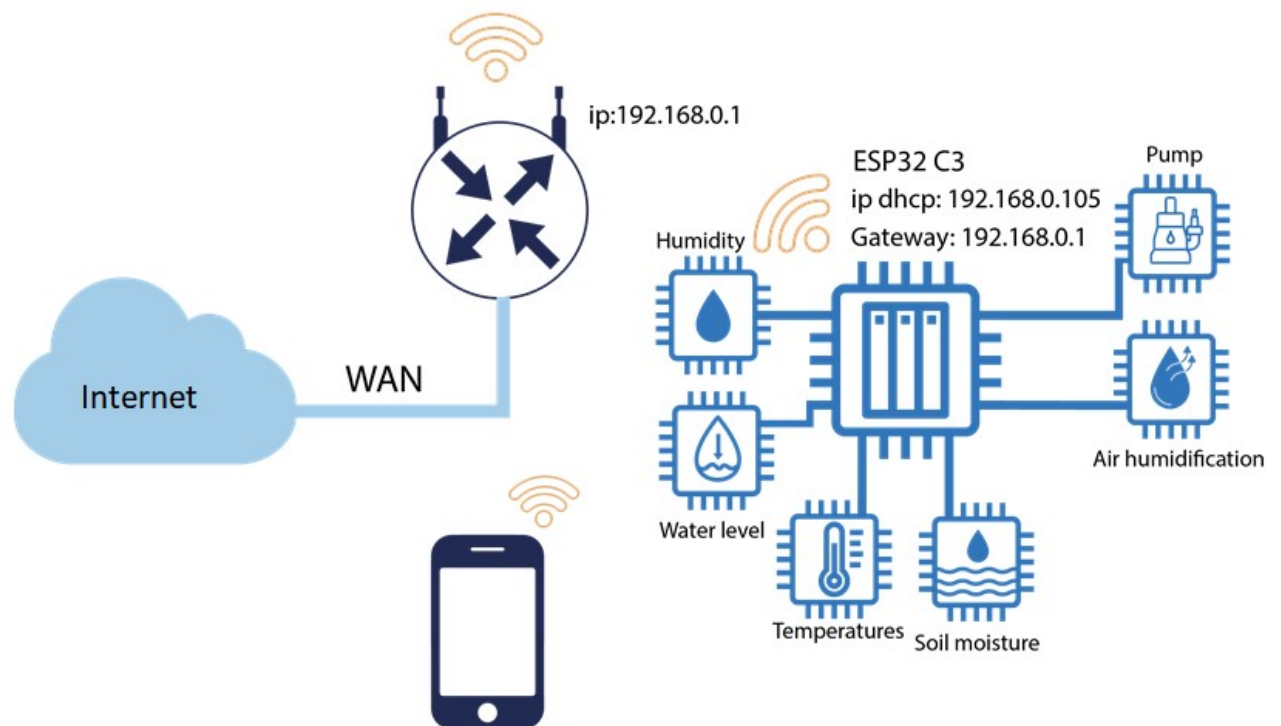


Рисунок 2.1 – Схема взаємодії елементів системи

Дана схема включає різноманітні компоненти для збору та передачі даних. Мікроконтролер системи – це ESP32 C3 з IP-адресою 192.168.0.105 і шлюзом 192.168.0.1. До ESP32 підключено датчі вологості ґрунту, вологості повітря, температури, рівня води та насос для поливу. Мікроконтролер ESP32 C3 отримує IP-адресу по DHCP з маршрутизатора. Для того щоб IP -адреса мікроконтролера залишалась незмінною на маршрутизаторі здійснено прив'язку IP, яку видає сервер DHCP до MAC адреси ESP32 C3.

Додатковим компонентом є смартфон, який підключений до маршрутизатора через Wi-Fi, що дає можливість отримання та надсилання даних на контролер.

Система підключена до мережі Інтернет через маршрутизатор з IP-адресою LAN 192.168.0.1. Схема показує взаємодію між різними компонентами, що забезпечують контроль клімату та вологості ґрунту в домашній теплиці.

2.2 Опис та характеристика елементів автоматизованої теплиці

2.2.1. Мікроконтролер ESP32-C3

На рисунку 2.2 показано мікроконтролер ESP32-C3, розроблений Espressif Systems, який забезпечує розширені можливості для вбудованих систем, зокрема в областях IoT. Ядро мікроконтролера RISC-V відрізняється низьким енергоспоживанням, що робить його ідеальним рішенням для розробки енергоефективних і компактних пристроїв. ESP32-C3 підтримує стандарти бездротового зв'язку, такі як Wi-Fi (IEEE 802.11 b/g/n) і Bluetooth Low Energy (BLE), що дозволяє створювати з'єднання між пристроями без додаткових зовнішніх компонентів [22].

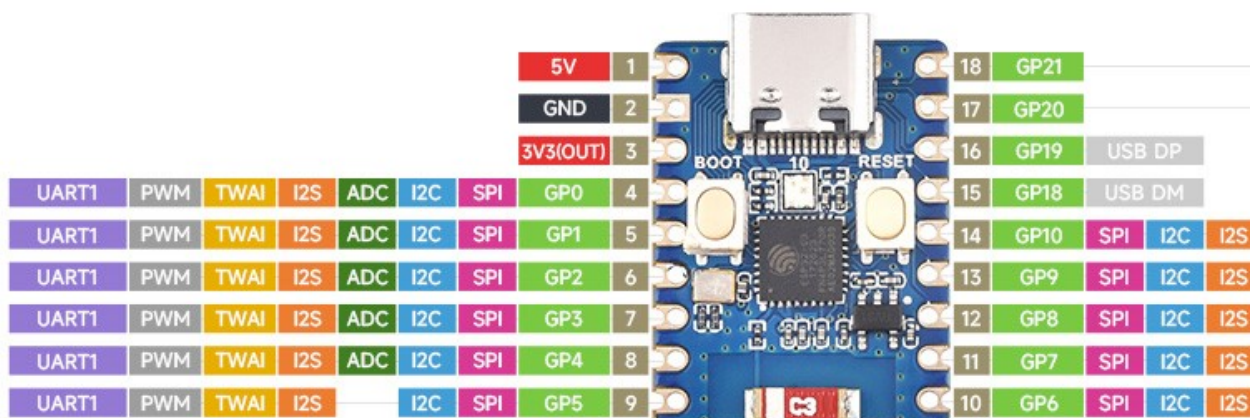


Рисунок 2.2 – Зовнішній вигляд та піни мікроконтролера ESP32 C3

До основних переваг ESP32-C3 належить його модульність та багатофункціональність, які відкривають широкі можливості для програмістів і інженерів, дозволяючи імплементувати різноманітні функції на одному чіпі. Це не тільки спрощує процес розробки, але й знижує загальну вартість виробництва пристроїв. Мікроконтролер оснащений рядом апаратних інтерфейсів, включаючи GPIO, ADC, UART, SPI, I2C, що робить його здатним до швидкої інтеграції з іншими електронними компонентами і системами.

З точки зору безпеки, ESP32-C3 має розширені засоби захисту даних і з'єднань, включаючи підтримку криптографічних алгоритмів, таких як AES, RSA, і ECC, а також можливість реалізації захищеного завантаження і шифрування пам'яті [23]. Це гарантує, що пристрої на базі ESP32-C3 можуть бути захищені від зовнішніх загроз і витоку інформації, що є критично важливим для пристроїв IoT.

На рисунку 2.3 показано функціональну блок-схему ESP32-C3, яка демонструє внутрішню архітектуру мікроконтролера, що включає ряд спеціалізованих модулів. Основою системи є 32-бітний RISC-V мікропроцесор, який сприяє ефективному обробленню даних з високою швидкістю. Цей процесор підтримується кеш-пам'яттю та SRAM, що забезпечує швидкий доступ до часто використовуваних даних, а також ROM для зберігання необхідного програмного забезпечення.

Радіочастотна (RF) секція містить балун з перемикачем для 2.4 ГГц діапазону, що дозволяє чіпу ефективно переключатися між прийомом та передачею сигналів, тим самим оптимізуючи використання енергії та зменшуючи загальне споживання ресурсів. Також присутній фазовий замок для стабілізації частоти, що є критично важливим для підтримання якості зв'язку.

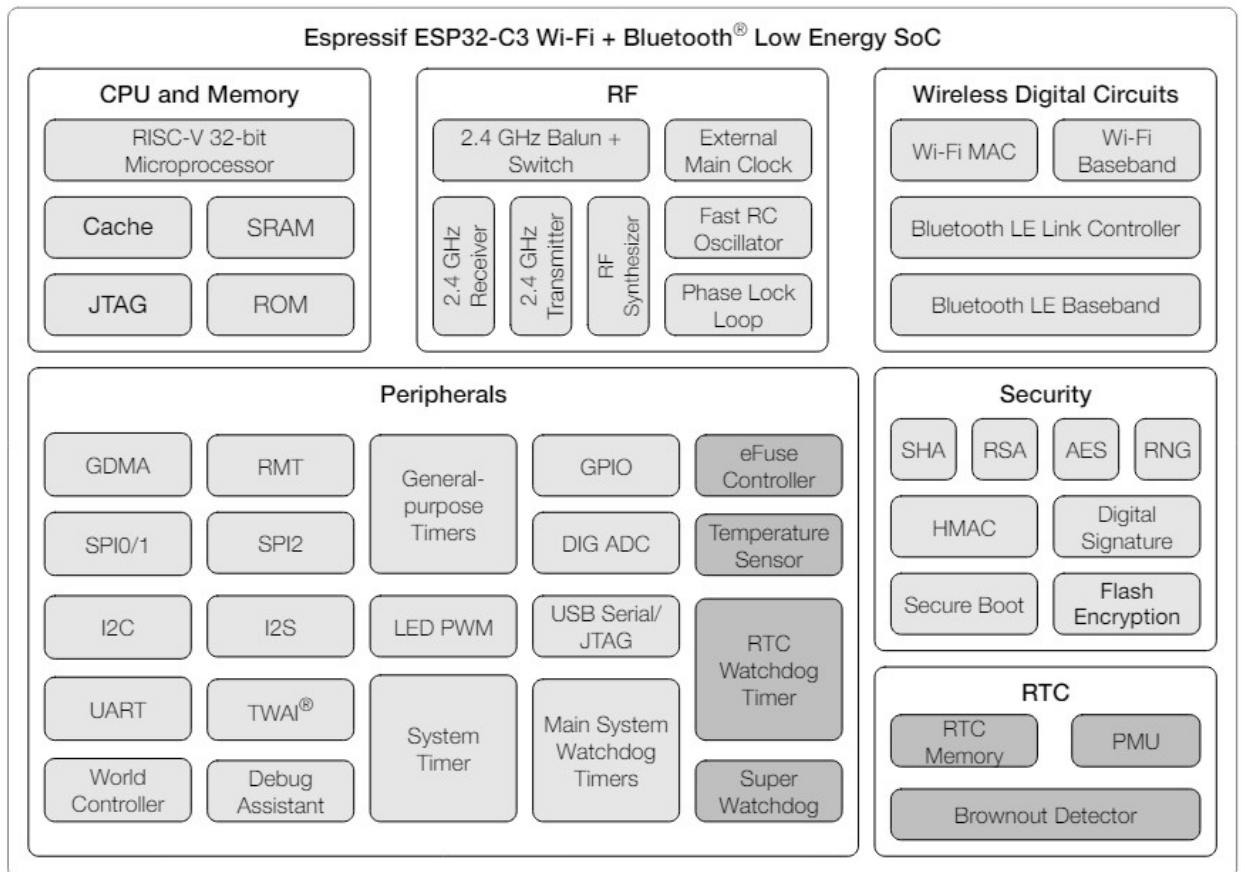


Рисунок 2.3 – Функціональна блок-схема ESP32 C3

Бездротові цифрові схеми обладнані модулями Wi-Fi та Bluetooth Low Energy, які забезпечують широкі можливості для мережеских з'єднань, включаючи зв'язок з Інтернет та між пристроями. Ці модулі використовуються для реалізації зв'язку з низьким енергоспоживанням та високою ефективністю, важливих для портативних та вбудованих пристроїв.

На додаток, система безпеки включає модулі для шифрування, автентифікації, та забезпечення цілісності даних, використовуючи сучасні криптографічні алгоритми такі як SHA, RSA, AES. Таке оснащення забезпечує високий рівень захисту від зовнішніх загроз, що є невід'ємною частиною роботи сучасних IoT систем. Secure Boot і Flash Encryption забезпечують додатковий рівень безпеки при завантаженні та зберіганні програмного забезпечення.

Останнім, але не менш важливим, є компоненти системи реального часу, що включають годинник реального часу і виявлення відмови живлення. Ці

функції дозволяють ESP32-C3 підтримувати точне відлік часу та швидко реагувати на зміни в стані живлення, що підвищує надійність системи в умовах нестабільного живлення.

2.2.2. Давач температури та вологості DHT11

На рисунку 2.4 показано DHT11, який являє собою компактний цифровий датчик, який використовується для вимірювання та моніторингу температури та вологості навколишнього середовища [24]. Цей датчик ідеально підходить для широкого спектру застосувань, від побутових до промислових, завдяки своїй надійності, економічності та простоті інтеграції з електронними системами.

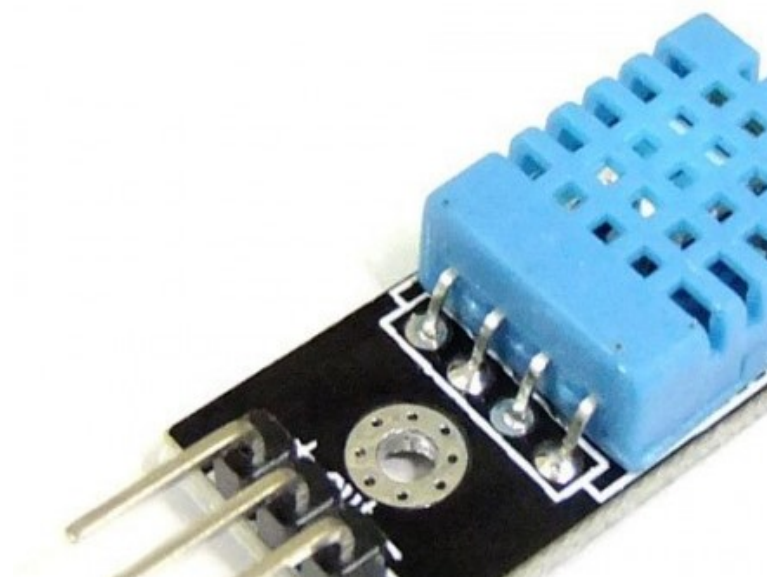


Рисунок 2.4 – Зовнішній вигляд датчика DHT22

Основною особливістю DHT11 є його здатність вимірювати відносну вологість повітря з точністю до 5% та температуру з точністю до 2°C у діапазоні від 0 до 50°C. Датчик використовує внутрішній термістор для вимірювання температури та ємнісний вологомір для визначення вологості, що дозволяє досягати високої точності вимірювань при мінімальному енергоспоживанні.

DHT11 простий у використанні і потребує лише одного цифрового піна для зв'язку з мікроконтролером, таким як Arduino або ESP. Він використовує простий протокол для передачі даних, що забезпечує зручність підключення та знижує витрати на додаткову проводку в складних системах.

DHT11 має малі розміри та може бути інтегрований у різноманітні пристрої без зайвих витрат на простір, що робить його ідеальним для застосувань, де простір обмежений, наприклад, у розумних будинках, портативних метеостанціях або системах контролю клімату. Його низька вартість також робить цей датчик привабливим вибором для студентів, які працюють над DIY-проектами та експериментами в галузі автоматизації дому та збору даних про навколишнє середовище.

2.2.3. Гігрометр FC-28 та компаратор напруги LM393

На рисунку 2.5 показано датчик вологості ґрунту FC-28, який широко застосовується у сфері сільського господарства, садівництва та робототехніки, завдяки його здатності точно вимірювати рівень вологості в ґрунті. Цей датчик використовує принцип електричного опору для оцінки вологості, вимірюючи провідність ґрунту, яка безпосередньо залежить від кількості вологості у ньому [25].

Конструкція FC-28 включає дві металеві зонди, які вводяться в ґрунт, і модуль управління, що аналізує отримані вимірювання. Через простоту свого механізму, FC-28 може швидко відгукнутися на зміни в рівні вологості, що робить його незамінним інструментом для систем автоматизованого поливу, де необхідний швидкий відгук на зміни умов довкілля.

Однією з ключових переваг FC-28 є його універсальність і простота інтеграції з різними мікроконтролерами, включаючи Arduino, ESP та Raspberry Pi, завдяки наявності аналогових та цифрових виходів. Цифровий вихід дозволяє датчику працювати як простий перемикач, що спрацьовує коли рівень вологості перевищує певний поріг, тоді як аналоговий вихід надає більш

детальну інформацію, що може бути використана для точнішого контролю вологості ґрунту.

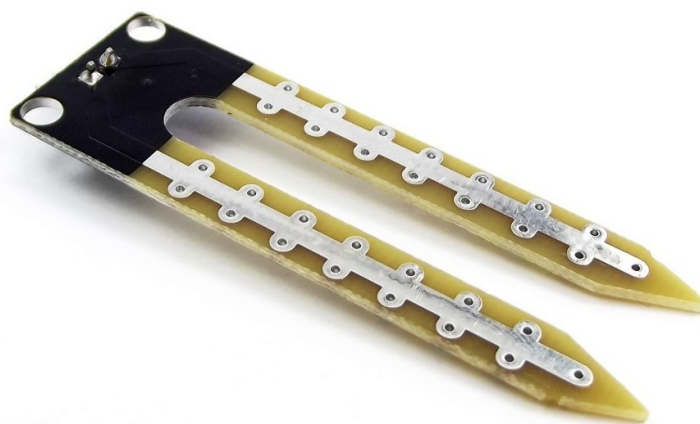


Рисунок 2.5 – Зовнішній вигляд давача FC-28

Для захисту від корозії і подовження терміну служби, зонди FC-28 зазвичай покриваються нікелем або іншим корозійно-стійким матеріалом, оскільки вони постійно контактують з вологим ґрунтом. Незважаючи на свою простоту, FC-28 вимагає регулярного очищення і обслуговування зондів для запобігання накопиченню мінералів і забезпечення точності вимірювань.

На рисунку 2.6 показано LM393, який є популярним і широко використовуваним двоканальним компаратором напруги, який займає важливе місце в електронному обладнанні завдяки своїй здатності швидко порівнювати різні напруги. Цей інтегральний мікросхемний компонент відомий своєю надійністю та ефективністю в широкому спектрі застосувань, включаючи контрольні системи, прості аналогові перемикачі та засоби автоматизації.

Конструкція LM393 базується на двох незалежних компараторах з відкритим колектором на виході, що дозволяє їм працювати з різними логічними рівнями і навантаженнями. Це особливо важливо у системах, де потрібно інтегрувати сигнали з різними амплітудами. LM393 може працювати в діапазоні напруги живлення від 2 до 36 вольт, що підходить для використання як з низьковольтними, так і з високовольтними джерелами живлення.

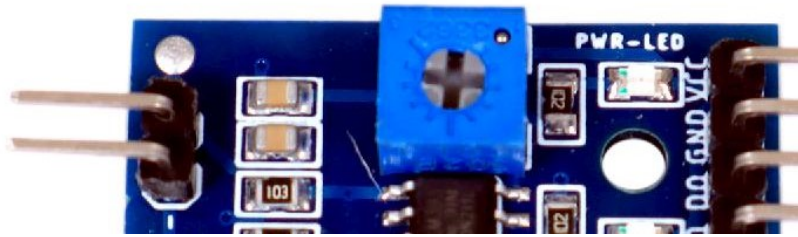


Рисунок 2.6 – Компаратора напруги LM393

Додатковою перевагою LM393 є його висока швидкість відгуку, що робить цей компонент ідеальним для використання в критичних за часом застосуваннях, де швидке виявлення змін напруги є вирішальним. Компаратор забезпечує точне порівняння вхідних сигналів, що дозволяє системам швидко реагувати на зміни та приймати рішення про подальші дії, засновані на порівнянні рівнів напруги.

У застосуваннях, де потрібна вища точність, LM393 демонструє відмінні характеристики завдяки низькому вхідному зсуву та стабільній роботі в широкому діапазоні температур. Його здатність працювати при різних температурних умовах робить його надійним вибором для зовнішніх застосувань і систем, які піддаються екстремальним умовам експлуатації.

2.2.4. Давач рівня води HW-038

На рисунку 2.7 показано давач рівня води HW-038, який є аналоговим модулем, який дозволяє вимірювати рівень води в контейнері за допомогою електричної провідності. В основі його роботи лежить принцип зміни електричного опору в залежності від контакту з водою. Коли рівень води піднімається, вода торкається низки паралельних проводів, що входять до складу давача, зменшуючи електричний опір і, відповідно, збільшуючи напругу сигналу [26].

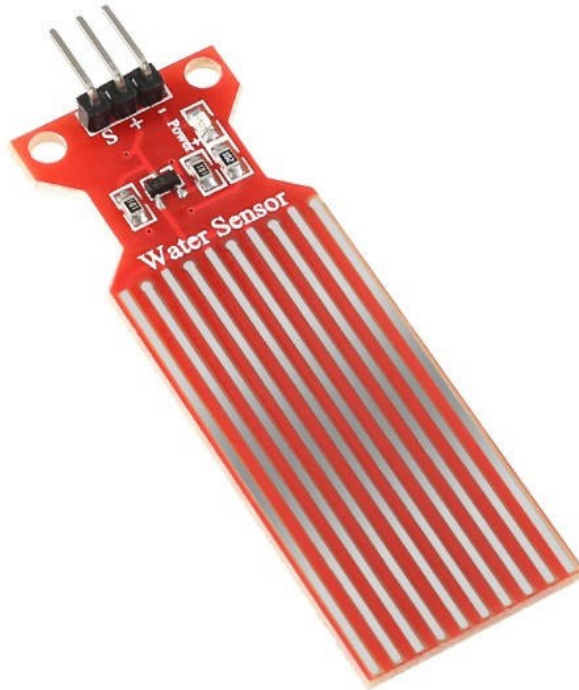


Рисунок 2.7 – Зовнішній вигляд давача HW-038

Модуль сумісний із мікроконтролерами, такими як Arduino та ESP, і має високу чутливість та низьке споживання енергії, що робить його ідеальним для застосування в пристроях, які живляться від батареї. Напруга виходу пропорційна рівню води: від 0 В, коли давач сухий, до напруги живлення, коли він повністю занурений у воду.

Залежно від напруги живлення, яка може коливатися між 3,3 В та 5 В, напруга виходу варіюється відповідно. Це важливо враховувати при проектуванні системи управління на базі давача HW-038.

Окрім технічних характеристик, таких як робоча напруга, струм, тип давача і діапазон робочих температур, HW-038 має такі особливості, як простота використання, точність вимірювань, універсальність, компактність та довговічність. Модуль можна використовувати як у приміщенні, так і на вулиці.

2.2.5. Водяний насос RS-360SH

На рисунку 2.8 показано мініатюрний водяний насос-помпа RS-360SH, який є компактним електромеханічним пристроєм, який призначений для переміщення води через малі системи, такі як домашні фонтани, акваріуми, а також у системи автоматичного поливу. Цей насос може похвалитися високою продуктивністю у порівняно малому корпусі, що робить його ідеальним для застосувань, де простір обмежений [27].



Рисунок 2.8 – Зовнішній вигляд водяного насоса–помпи RS-360SH

RS-360SH функціонує на постійному струмі та має низьке енергоспоживання, що забезпечує економічну та ефективну роботу в умовах, що вимагають постійної роботи пристрою. Насос оснащений мотором постійного струму, який виконує обертання ротора, створюючи тим самим необхідний напір для переміщення води.

Однією з ключових характеристик RS-360SH є його здатність створювати значний тиск при низьких рівнях шуму, що робить його використання комфортним навіть у житлових умовах. Крім того, насос має високу надійність та довговічність, що вирізняє його серед аналогічних пристроїв у своєму класі.

Застосування помпи RS-360SH не обмежується лише невеликими водоймами або акваріумами. Вона також може використовуватися у портативних охолоджувачах, системах циркуляції води в промислових умовах та в інших технічних системах, де потрібне мінімальне споживання енергії та надійність роботи..

2.2.6. Транзисторний модуль MOSFET IRF520N

На рисунку 2.9 показано транзисторний модуль MOSFET IRF520N, розроблений компанією International Rectifier, що є частиною їхньої п'ятої генерації HEXFET транзисторів. Використання передових технологій обробки дозволяє досягти надзвичайно низького опору в режимі відкритого каналу на одиницю площі кремнію [28]. Цей прогрес у технології сприяє високій ефективності перетворення, що, в свою чергу, разом із швидкістю комутації та міцністю пристрою, робить IRF520 ідеальним вибором для широкого спектра застосувань.

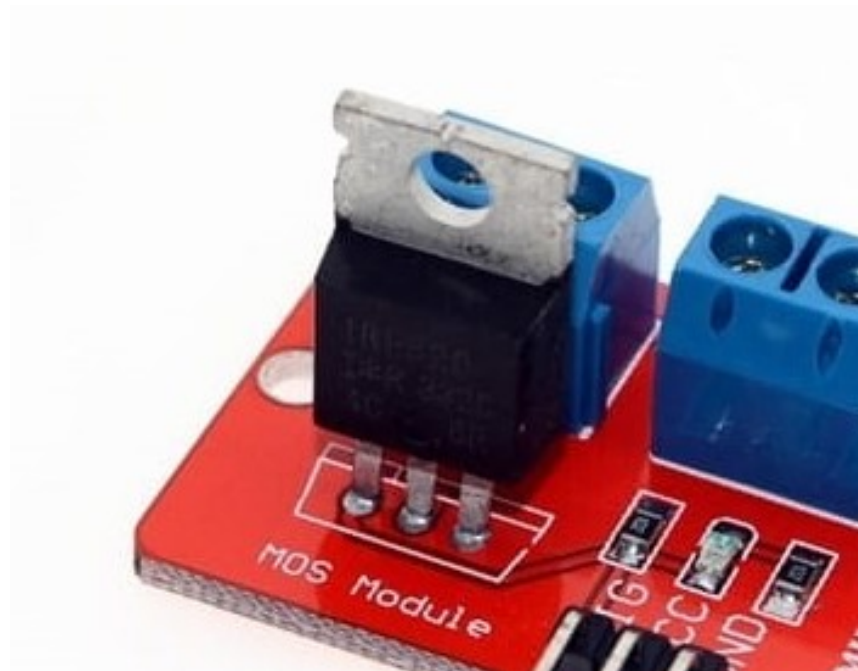


Рисунок 2.9 – Зовнішній вигляд транзисторного модуля

Транзистор вмонтований у корпус TO-220, який є стандартним вибором для комерційних та промислових застосувань при рівні випромінювання тепла до приблизно 50 ватт. Низький тепловий опір та вартість корпусу сприяють його популярності в індустрії.

IRF520 забезпечує постійний струм стоку до 9.7 А при температурі корпусу 25°C із напругою управління 10 В. При температурі корпусу 100°C максимальний струм стоку знижується до 6.8 А. Транзистор також здатен витримувати імпульсний струм до 38 А та має максимальну потужність розсіювання 48 Вт при 25°C.

Що стосується надійності, IRF520 може працювати при температурах від -55 до +175°C, що робить його придатним для використання в умовах високої температури.

2.2.7. Водонепроникний корпус SONOFF

На рисунку 2.10 показано корпус SONOFF Waterproof Box IP66. Це водонепроникний захисний корпус, призначений для надійного захисту електричного обладнання в умовах впливу вологи, пилу та бризок. Цей корпус відповідає міжнародному стандарту IP66, що забезпечує високий рівень захисту від води та пилу, дозволяючи обладнанню функціонувати ефективно навіть у важких зовнішніх умовах [29].

Цей корпус виготовлений із високоміцного полімерного матеріалу, який забезпечує стійкість до ударів, корозії, ультрафіолету та температурних коливань. Його міцна конструкція включає ущільнення та спеціальні клапани для вентиляції, які запобігають конденсації всередині корпусу, тим самим захищаючи вміст від вологи та забезпечуючи тривалий термін служби електронних компонентів.



Рисунок 2.10 – Зовнішній вигляд корпусу SONOFF

Конструкція корпусу також передбачає легкий доступ до вмісту для обслуговування та налаштування. Це забезпечує зручність в експлуатації та мінімізує ризик пошкодження компонентів під час доступу до них.

SONOFF Waterproof Box IP66 має універсальність використання і підходить не тільки для захисту стаціонарного обладнання, але й для мобільних рішень, де необхідний захист від води та пилу. Його використання знайшло застосування в домашній автоматизації, на виробництвах, у сільському господарстві, а також для зовнішнього освітлення та інших зовнішніх інсталяцій.

2.2.8. Ультразвуковий зволожувач повітря

На рисунку 2.11 показано компактний ультразвуковий зволожувач повітря, що є гарним варіантом в області покращення мікроклімату приміщень. Основні компоненти цього зволожувача включають плату керування, фільтр, і мембрану ультразвукового випромінювача, що разом створюють ефективний механізм для перетворення води в мікроскопічний туман.



Рисунок 2.11 – Зовнішній вигляд ультразвукового зволожувача повітря

Центральним елементом є ультразвукова мембрана, яка, коли опущена в ємність з водою, активується від підключення до блоку живлення через роз'єм Micro-USB. Під впливом ультразвукових хвиль, що генеруються на частоті приблизно 108-113 кГц, вода над мембраною перетворюється на водяний туман. Це дозволяє зволожувачу насичувати повітря мікроскопічними частинками води.

Користувачеві необхідно просто наповнити підходящу ємність, таку як стакан чи чашку, подати вологу на мембрану та підключити зволожувач до джерела живлення. Завдяки своїм компактним розмірам плата керування легко інтегрується у домашнє обладнання, а її невелика потужність споживання в 2 Вт і струм 0.3 А роблять цей зволожувач дуже економічним у використанні.

2.3 Електрична принципова схема системи

На рисунку 2.12 зображена електрична принципова схема системи контролю клімату та вологості ґрунту. В даній схемі мікроконтролер ESP32 C3 використовується як центральний вузол управління, до якого підключено різноманітні датчі та пристрої для забезпечення автоматизованого моніторингу та контролю умов середовища.

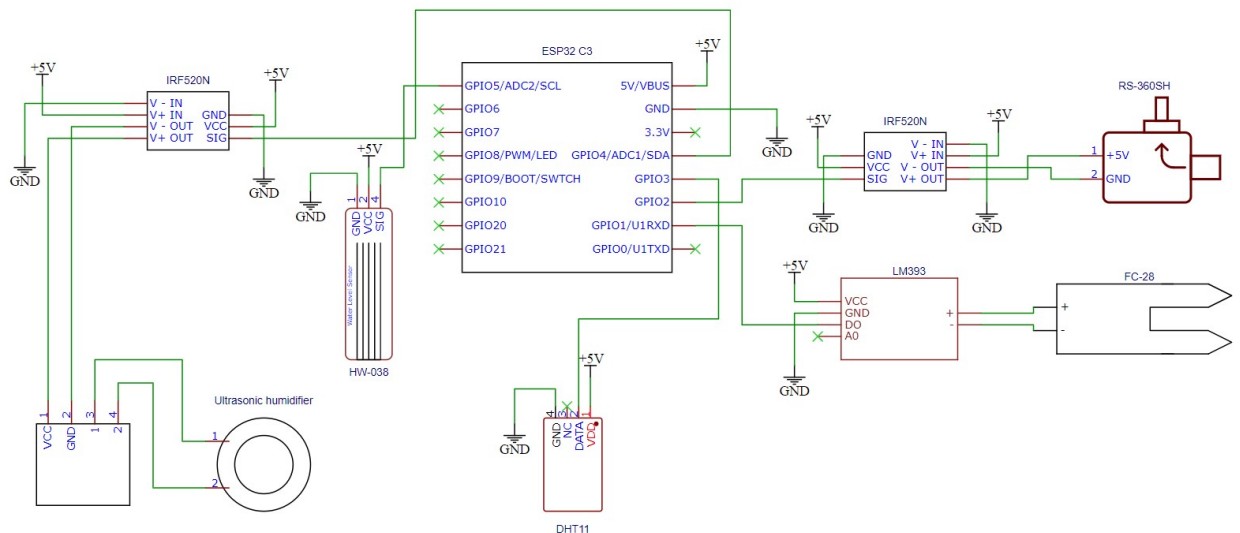


Рисунок 2.12 – Електрична принципова схема системи

ESP32 C3 забезпечує високу гнучкість та можливість швидкої обробки даних, що надходять від підключених компонентів, а також дозволяє використовувати різні протоколи зв'язку для взаємодії з додатковим обладнанням.

Гігрометр FC-28, який підключено через компаратор напруги LM393, до GPIO1, передає дані про вологість ґрунту безпосередньо до мікроконтролера, що дозволяє системі миттєво реагувати на зміни вологості. Додатково, за допомогою IRF520N, який підключено до GPIO2, реалізовано управління водяним насосом. Цей елемент відіграє ключову роль у забезпеченні подання води до ґрунту за потреби, враховуючи дані від гігрометра та інших датчиків.

Датчик температури та вологості DHT11, підключений до GPIO3, дозволяє системі отримувати точні показники температури повітря, що є важливим для забезпечення оптимальних умов. IRF520N, з'єднаний з GPIO4, використовується для управління ультразвуковим зволожувачем повітря, що забезпечує необхідний рівень вологості в середовищі контролю, що також залежить від показників від DHT11.

Ключовим елементом для контролю за ресурсами системи є датчик рівня води, який підключено до GPIO5. Він дозволяє системі моніторити кількість

доступної води, що важливо для підтримки безперервної роботи водяного насоса та зволожувача повітря.

2.4 Висновок до другого розділу

У другому розділі детально описано елементи контролю клімату та вологості ґрунту для домашньої теплиці. Показано інтеграцію різноманітних компонентів для створення автоматизованого моніторингу та управління кліматичними умовами. Важливою складовою системи є мікроконтролер ESP32 С3, який використовується як центральний вузол управління, до якого підключаються датчики для вимірювання температури, вологості повітря, вологості ґрунту та рівня води.

Важливу роль в системі відіграють датчики та датчики, зокрема DHT11 та FC-28, які забезпечують контроль за умовами у теплиці. Ці дані є основою для автоматичного регулювання таких систем, як полив або зволоження повітря, з використанням водяних насосів та ультразвукових зволожувачів. Інтеграція транзисторного модуля MOSFET IRF520N дозволяє ефективно управляти потужнішими виконавчими механізмами системи, забезпечуючи надійність та ефективність роботи.

РОЗДІЛ 3. НАЛАШТУВАННЯ ТА ТЕСТУВАННЯ СИСТЕМИ

3.1 Вибір мови програмування для ESP32 C3

Одним із найважливіших аспектів розробки інтерактивних електронних пристроїв є вибір мови програмування та відповідного програмного забезпечення. У випадку з ESP, мова базується на C/C++, що робить її потужним інструментом для програмування мікроконтролерів. Використання саме цієї мови обумовлене декількома ключовими факторами, які забезпечують ефективність та доступність для широкого кола користувачів [30].

C/C++ є однією з найпоширеніших мов програмування в галузі вбудованих систем. Це обумовлено її високою продуктивністю, гнучкістю та здатністю ефективно працювати з апаратним забезпеченням. Мова C має багатий набір інструментів для роботи з пам'яттю, що дозволяє максимально ефективно використовувати обмежені ресурси мікроконтролерів. C++ додає до цього об'єктно-орієнтовані можливості, які допомагають структурувати код та спрощують розробку складних систем. Тому вибір C/C++ як базової мови для програмування ESP є логічним та обґрунтованим рішенням.

Щодо програмного забезпечення, Arduino IDE (Integrated Development Environment) є також середовищем розробки для платформ ESP. Воно було створено спеціально для того, щоб забезпечити простий та інтуїтивно зрозумілий інтерфейс для написання, редагування та завантаження коду на мікроконтролерні плати. Основною перевагою Arduino IDE є його доступність та легкість у використанні, що робить його ідеальним вибором для новачків та ентузіастів, які тільки починають свій шлях у світі електроніки та програмування.

Arduino IDE підтримує різні моделі плат ESP, що дозволяє легко перемикатися між різними апаратними платформами без необхідності змінювати програмне забезпечення. Це досягається за рахунок використання бібліотек, які абстрагують складність роботи з апаратною частиною, надаючи

прості у використанні функції для взаємодії з різними компонентами, такими як датчики, двигуни, світлодіоди та інші. Бібліотеки спрощують процес розробки, дозволяючи зосередитися на логіці роботи пристрою, а не на низькорівневих деталях.

Ще однією важливою особливістю Arduino IDE є його кросплатформність. Середовище розробки доступне для операційних систем Windows, macOS та Linux, що забезпечує широку сумісність та можливість працювати на будь-якому комп'ютері. Це особливо важливо для освітніх закладів та лабораторій, де використовуються різні операційні системи.

Arduino IDE має вбудовану підтримку компіляції та завантаження коду на плату, що робить процес розробки швидким та безперервним. Крім того, інтегроване середовище має вбудований послідовний порт моніторингу, який дозволяє здійснювати відлагодження та моніторинг роботи пристрою в реальному часі. Це є важливим інструментом для розробників, оскільки дозволяє швидко знаходити та виправляти помилки, а також тестувати різні функції пристрою.

Arduino IDE підтримує розширення через додаткові бібліотеки та плагіни, що дозволяє розробникам додавати нові функціональні можливості та інтегрувати середовище з іншими інструментами. Наприклад, існують бібліотеки для роботи з Wi-Fi, Bluetooth, різними датчиками та іншими модулями, що значно розширює можливості розробки. Крім того, спільнота Arduino активно розробляє та підтримує нові бібліотеки, що робить екосистему фреймворку постійно зростаючою та еволюціонуючою.

3.2 Опис програмного коду для ESP32 C3

Програмування плати ESP32 виконувалось у середовищі розробки Arduino IDE, що дозволило створювати код для контролю над різними електронними компонентами. Програмний код містить дві основні функції: `setup()` та `loop()`. Функція `setup()` виконується один раз при старті програми і

зазвичай служить для ініціалізації параметрів, таких як налаштування режимів роботи пінів або ініціалізація серійного порту. Функція `loop()` виконується циклічно і містить основну логіку програми, що дозволяє платі виконувати повторювані операції.

Фрагмент коду, який ініціалізує бібліотеки та підключає різні датчики до ESP32 C3 (див. лістинг 3.1).

Лістинг 3.1 – Підключення бібліотек та датчиків

```
include <WiFi.h>
#include <ESPAsyncWebServer.h>
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <Preferences.h>
#define SOIL_MOISTURE_PIN 1
#define WATER_PUMP_PIN 2
#define HUMIDITY_SENSOR_PIN 3
#define FAN_PIN 4
#define WATER_LEVEL_PIN 5
int soilMoistureThreshold = 300;
int humidityThreshold = 75;
DHT dht(HUMIDITY_SENSOR_PIN, DHT11);
```

На початку коду підключаються необхідні бібліотеки, включаючи `WiFi` для підключення до бездротової мережі, `ESPAsyncWebServer` для створення веб-сервера, `Wire` для роботи з шиною I2C, `Adafruit_Sensor` та `DHT` для роботи з датчиком вологості та температури, `Preferences` для зберігання налаштувань у пам'яті. Визначаються пін-коди для підключення датчиків та виконавчих елементів. Наприклад, `SOIL_MOISTURE_PIN` використовується для датчика вологості ґрунту, `WATER_PUMP_PIN` для керування водяним насосом, `HUMIDITY_SENSOR_PIN` для датчика вологості повітря, `FAN_PIN` для зволожувача повітря, а `WATER_LEVEL_PIN` для датчика рівня води. Також

визначаються початкові порогові значення для вологості ґрунту та повітря, які можуть бути змінені користувачем через веб-інтерфейс. Ініціалізується об'єкт `dht` для роботи з давачем вологості та температури.

У наведеному фрагменті коду, оголошуються змінні, які будуть використовуватись для зберігання поточних значень вологості ґрунту, вологості повітря та рівня води. (див. лістинг 3.2).

Лістинг 3.2 – Ініціалізація змінних та підключення до Wi-Fi

```
int soilMoistureValue = 0;
float humidity = 0.0;
int waterLevelValue = 0;
const char* ssid = "SSID";
const char* password = "XXXXXXXX";
AsyncWebServer server(80);
Preferences preferences;
```

Змінні `soilMoistureValue`, `humidity` та `waterLevelValue` використовуються для зберігання значень, зчитаних з відповідних давачів. Задаються SSID та пароль для підключення до Wi-Fi мережі, що дозволяє платі ESP32 підключатися до Інтернет. Створюється об'єкт `AsyncWebServer` для налаштування веб-сервера, який буде слухати на порту 80. Ініціалізується об'єкт `preferences` для роботи з пам'яттю, де будуть зберігатися налаштування, такі як порогові значення вологості.

В лістингу 3.3 показано фрагмент коду функції `setup()`, що відповідає за ініціалізацію давачів, налаштування пінів, підключення до Wi-Fi та зчитування збережених порогових значень для вологості ґрунту та вологості повітря

Лістинг 3.3 – Ініціалізація компонентів у функції `setup`

```
void setup() {
  Serial.begin(115200);
  pinMode(SOIL_MOISTURE_PIN, INPUT);
  pinMode(WATER_PUMP_PIN, OUTPUT);
```

```
pinMode(FAN_PIN, OUTPUT);
pinMode(WATER_LEVEL_PIN, INPUT);
dht.begin();
```

Функція `setup()` виконується один раз при старті програми. Спочатку ініціалізується послідовний зв'язок з швидкістю 115200 біт/с для виведення діагностичної інформації. Далі задаються режими роботи пінів: пін для давача вологості ґрунту налаштовується як вхідний, пін для водяного насоса як вихідний, пін для давача рівня води як вхідний. Викликається метод `begin()` для ініціалізації давача вологості та температури `dht`.

Фрагмент коду, що відповідає за підключення плати до Wi-Fi мережі показано на лістингу 3.4.

Лістинг 3.4 – Підключення до Wi-Fi

```
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
}
Serial.println("Connected to WiFi");
Serial.print("IP Address: ");
Serial.println(WiFi.localIP());
```

Метод `WiFi.begin()` ініціалізує процес підключення до мережі, використовуючи задані SSID та пароль. Поки стан з'єднання не буде змінено на підключений (`WL_CONNECTED`), програма буде чекати з інтервалом у 1000 мілісекунд і виводити повідомлення "Connecting to WiFi...". Після успішного підключення до мережі, виводиться повідомлення "Connected to WiFi" та IP-адреса, яку отримала плата.

Фрагмент коду, що зчитує порогові значення вологості ґрунту та повітря з пам'яті показано в лістингу 3.5.

Лістинг 3.5 – Зчитування порогових значень з пам'яті

```

preferences.begin("settings", false);
    soilMoistureThreshold    =    preferences.getInt("soilMoisture",
soilMoistureThreshold);
    humidityThreshold        =    preferences.getInt("humidity",
humidityThreshold);
    preferences.end();
}

```

Метод `preferences.begin()` відкриває область пам'яті з назвою "settings" у режимі читання і запису. Викликаючи методи `getInt()`, програма отримує збережені значення порогів для вологості ґрунту та повітря. Якщо значення не збереглися, використовується стандартне значення, задане раніше в коді. Метод `preferences.end()` закриває доступ до пам'яті. Це забезпечує збереження налаштувань навіть після перезавантаження плати.

Фрагмент коду в лістингу 3.6 створює обробник для GET-запиту на маршруті "/", що дозволяє користувачам переглядати поточні значення давача та налаштовувати порогові значення через веб-інтерфейс. Це один із найважливіших елементів програми, оскільки він забезпечує взаємодію користувача з системою через веб-браузер.

Лістинг 3.6 – Код створення основної сторінки веб-сервера

```

server.on("/", HTTP_GET, [] (AsyncWebServerRequest *request) {
    String html = "<html><head><style>";
    html += "body { font-family: Arial, sans-serif; background-
color: #e6ffe6; }";
    html += ".container { max-width: 600px; margin: auto; padding:
20px; background-color: #ffffff; box-shadow: 0 0 10px rgba(0, 0,
0, 0.1); }";
    html += "h1 { color: #4CAF50; }";
    html += "p { font-size: 18px; }";
    html += "form { margin-top: 20px; }";
    html += "label { display: block; margin-bottom: 10px; }";
}

```

```

    html += "input[type='number'] { width: 100%; padding: 8px;
margin-bottom: 20px; border: 1px solid #ccc; border-radius: 4px;
}";

    html += "input[type='submit'] { background-color: #4CAF50;
color: white; padding: 10px 20px; border: none; border-radius:
4px; cursor: pointer; }";

    html += "input[type='submit']:hover { background-color:
#45a049; }";

    html += "</style></head><body>";
    html += "<div class='container'>";
    html += "<h1>ESP32 Sensor Data</h1>";
    html += "<p>Soil Moisture: " + String(soilMoistureValue) +
"</p>";

    html += "<p>Humidity: " + String(humidity) + "</p>";
    html += "<p>Water Level: " + String(waterLevelValue) + "</p>";
    html += "<form action='/set' method='POST'>";
    html += "    <label for='soilMoisture'>Soil Moisture
Threshold:</label>";
    html += "    <input type='number' id='soilMoisture'
name='soilMoisture' value='" + String(soilMoistureThreshold) +
"'>";
    html += "    <label for='humidity'>Humidity Threshold:</label>";
    html += "    <input type='number' id='humidity' name='humidity'
value='" + String(humidityThreshold) + "'>";
    html += "    <input type='submit' value='Set Thresholds'>";
    html += "</form>";
    html += "</div></body></html>";
    request->send(200, "text/html", html);
});

```

В даному коді використовується бібліотека ESPAsyncWebServer, яка забезпечує асинхронну обробку веб-запитів, що дозволяє створювати більш швидкодійні та відзивні веб-додатки на базі платформи ESP32.

HTML-сторінка, яка відправляється користувачеві, створюється безпосередньо у тілі обробника запиту. Вона починається з відкриття HTML-тегів `<html><head><style>`, що дозволяє вбудувати CSS-стили безпосередньо в документ. Стили задають зовнішній вигляд сторінки, забезпечуючи приємний і зрозумілий інтерфейс для користувача. Вони визначають основні елементи сторінки, такі як фоновий колір, шрифти та розміщення контейнерів.

Фоновий колір сторінки встановлюється як світло-зелений (`#e6ffe6`), що створює спокійну атмосферу для перегляду інформації. Основний контейнер (`.container`) має максимальну ширину 600 пікселів, вирівнюється по центру і має відступи по 20 пікселів з кожного боку. Це робить сторінку адаптивною і зручною для перегляду на різних пристроях. Контейнер також має білий фон і тінь, що додає візуального відокремлення від фону.

Заголовки (`h1`) встановлені з кольором `#4CAF50`, що надає яскравий і привабливий вигляд. Параграфи (`p`) мають збільшений розмір шрифту (18 пікселів), що полегшує читання тексту. Форма для введення даних (`form`) має відступ зверху 20 пікселів, що відокремлює її від інших елементів сторінки, роблячи інтерфейс більш зрозумілим.

Кожен елемент форми, такий як мітки (`label`) і поля введення (`input[type='number']`), має стилі, що забезпечують зручність використання. Мітки відображаються блоками з відступами знизу 10 пікселів, що робить їх легко видимими і зрозумілими. Поля введення мають ширину 100%, що дозволяє їм займати всю доступну ширину контейнера, відступи, внутрішні відступи та межі, що робить їх зручними для взаємодії.

Кнопки відправки (`input[type='submit']`) мають зелений фон (`#4CAF50`), білий текст і заокруглені краї, що робить їх привабливими і легко впізнаваними. При наведенні на кнопку змінюється фон на трохи темніший зелений (`#45a049`), що надає візуальний зворотний зв'язок користувачеві.

Тіло сторінки (`<body>`) містить основний контейнер, всередині якого розташовані заголовки (`<h1>`), параграфи з поточними значеннями датчиків і форма для налаштування порогових значень. Заголовок "ESP32 Sensor Data"

інформує користувача про призначення сторінки. Параграфи відображають поточні значення вологості ґрунту, вологості повітря та рівня води, використовуючи змінні `soilMoistureValue`, `humidity` та `waterLevelValue`, перетворені на рядки за допомогою функції `String()`.

Форма для налаштування порогових значень містить два поля введення для встановлення нових порогових значень вологості ґрунту та повітря. Поля введення заповнюються поточними значеннями порогів, зчитаними з відповідних змінних `soilMoistureThreshold` та `humidityThreshold`. Це дозволяє користувачу бачити поточні налаштування і, за потреби, змінювати їх.

Коли користувач натискає кнопку відправки, форма надсилає дані на сервер через POST-запит до маршруту `"/set"`. Це дозволяє оновити порогові значення і зберегти їх у пам'яті, щоб вони залишалися доступними після перезавантаження плати. Після відправки форми сервер обробляє запит і, за потреби, оновлює сторінку з новими налаштуваннями.

Заключний рядок коду `request->send(200, "text/html", html)`; відповідає за відправку HTML-сторінки користувачу. Параметр 200 означає, що запит був оброблений успішно, а рядок `"text/html"` вказує, що відправляється HTML-документ. Змінна `html`, що містить всю структуру сторінки, надсилається як вміст відповіді.

Фрагмент в лістингу 3.7 коду обробляє POST-запит для маршруту `"/set"`, який використовується для оновлення порогових значень вологості ґрунту та повітря.

Лістинг 3.7 – Обробка POST-запиту для оновлення порогових значень

```
server.on("/set", HTTP_POST, [] (AsyncWebServerRequest *request) {
    if (request->hasParam("soilMoisture", true) && request-
>hasParam("humidity", true)) {
        soilMoistureThreshold = request->getParam("soilMoisture",
true)->value().toInt();
        humidityThreshold = request->getParam("humidity", true)-
>value().toInt();
```

```

preferences.begin("settings", false);
preferences.putInt("soilMoisture", soilMoistureThreshold);
preferences.putInt("humidity", humidityThreshold);
preferences.end();

request->send(200, "text/html", "<html><body><h1>Settings
Updated</h1><p><a href='/'>Go back</a></p></body></html>");
} else {
    request->send(400, "text/html", "<html><body><h1>Invalid
Input</h1></body></html>");
}
});

```

Коли користувач відправляє форму з новими пороговими значеннями, веб-сервер перевіряє наявність цих значень у запиті. Якщо значення присутні, вони перетворюються на цілі числа та зберігаються у відповідних змінних. Потім ці нові значення записуються в пам'ять за допомогою об'єкта `preferences`, щоб зберегти їх навіть після перезавантаження плати. Після успішного оновлення налаштувань користувачу відправляється сторінка з повідомленням "Settings Updated". Якщо введені значення некоректні або відсутні, сервер відправляє сторінку з повідомленням "Invalid Input".

У цьому короткому, але важливому фрагменті коду ініціалізується веб-сервер (див. лістинг 3.8).

Лістинг 3.8 – Ініціалізація веб-сервера

```

server.begin();
}

```

Метод `server.begin()` запускає сервер та починає прослуховування запитів на заданому порту (в даному випадку, порт 80). Це дозволяє користувачам підключатися до сервера через веб-браузер і взаємодіяти з інтерфейсом для

перегляду даних з датчиків та оновлення порогових значень. Даний фрагмент завершує налаштування веб-сервера, після чого він готовий до обробки запитів.

Фрагмент коду в лістингу 3.9 виконує функцію `loop()` циклічно і містить основну логіку програми.

Лістинг 3.9 – Основний цикл програми (зчитування вологості ґрунту)

```
void loop() {
  soilMoistureValue = analogRead(SOIL_MOISTURE_PIN);
  Serial.print("Soil Moisture: ");
  Serial.println(soilMoistureValue);
}
```

Спочатку зчитується значення вологості ґрунту з відповідного піну за допомогою функції `analogRead()`. Отримане значення зберігається у змінну `soilMoistureValue`. Потім це значення виводиться на послідовний монітор для діагностичних цілей. Ця інформація дозволяє користувачеві спостерігати за рівнем вологості ґрунту в режимі реального часу, що є важливим для відлагодження.

Фрагмент коду в лістингу 3.10 відповідає за зчитування значення вологості повітря з датчика DHT11, який підключений до відповідного піну.

Лістинг 3.10 – Зчитування вологості повітря

```
humidity = dht.readHumidity();
Serial.print("Humidity: ");
Serial.println(humidity);
```

Метод `readHumidity()` викликається на об'єкті `dht`, що ініціалізований раніше у функції `setup()`. Це значення зберігається у змінній `humidity`, яка визначена на початку програми. Після зчитування значення вологості повітря воно виводиться на послідовний монітор за допомогою функцій `Serial.print()` та `Serial.println()`. Це дозволяє користувачеві бачити поточне значення вологості повітря у режимі реального часу, що може бути корисним для відлагодження.

Фрагмент коду відповідає за зчитування значення рівня води з відповідного датчика, підключеного до піну, визначеного як WATER_LEVEL_PIN (див. лістинг 3.11).

Лістинг 3.11 – Зчитування рівня води

```
waterLevelValue = analogRead(WATER_LEVEL_PIN);  
Serial.print("Water Level: ");  
Serial.println(waterLevelValue);
```

Метод `analogRead()` зчитує аналогове значення з цього піну і зберігає його у змінній `waterLevelValue`. Потім це значення виводиться на серійний монітор за допомогою функцій `Serial.print()` та `Serial.println()`. Це дозволяє користувачеві спостерігати за рівнем води в режимі реального часу, що може бути важливим для системи автоматичного поливу або для контролю рівня води у резервуарі.

Фрагмент коду в лістингу 3.12 відповідає за контроль роботи водяного насоса на основі зчитаного значення вологості ґрунту.

Лістинг 3.12 – Контроль водяного насоса

```
if (soilMoistureValue > soilMoistureThreshold) {  
    digitalWrite(WATER_PUMP_PIN, HIGH);  
    delay(3000);  
    digitalWrite(WATER_PUMP_PIN, LOW);  
}
```

Якщо значення вологості ґрунту (`soilMoistureValue`) перевищує встановлений поріг (`soilMoistureThreshold`), то водяний насос включається за допомогою встановлення піну `WATER_PUMP_PIN` у високий рівень (`HIGH`). Після цього виконується затримка на 3000 мілісекунд (3 секунди) за допомогою функції `delay()`, що дозволяє насосові працювати протягом цього часу. Після затримки пін встановлюється у низький рівень (`LOW`), що вимикає насос. Це

забезпечує автоматичне включення насоса для поливу рослин, коли вологість ґрунту стає занадто високою, запобігаючи перезволоженню.

Даний фрагмент коду відповідає за контроль зволожувача повітря на основі зчитаного значення вологості повітря (див. лістинг 3.13).

Лістинг 3.13 – Контроль зволожувача повітря

```
if (humidity > humidityThreshold) {  
    digitalWrite(FAN_PIN, HIGH);  
} else {  
    digitalWrite(FAN_PIN, LOW);  
}
```

Якщо значення вологості повітря (`humidity`) перевищує встановлений поріг (`humidityThreshold`), то зволожувач вимикається за допомогою встановлення піну `FAN_PIN` у високий рівень (`HIGH`). Якщо ж значення вологості не перевищує поріг, то зволожувач включається, встановлюючи пін у низький рівень (`LOW`). Це дозволяє автоматично підтримувати оптимальний рівень вологості повітря, забезпечуючи комфортні умови для рослин або інших об'єктів у контрольованому середовищі.

Даний фрагмент коду додає затримку на 2000 мілісекунд (2 секунди) перед початком наступного циклу функції `loop()` (див. лістинг 3.14).

Лістинг 3.14 – Затримка перед наступним циклом

```
delay(2000);  
}
```

Це дозволяє зменшити частоту зчитування значень з датчиків і виконання контрольних дій, що може бути корисним для зменшення навантаження на систему та підвищення стабільності роботи. Така затримка дає можливість системі відновлюватися між циклами та знижує ймовірність перевантаження датчиків або виконавчих пристроїв.

Цей фрагмент коду визначає функцію `saveThresholds()`, яка зберігає порогові значення вологості ґрунту та повітря у пам'ять (див. лістинг 3.15).

Лістинг 3.15 – Збереження порогових значень у пам'ять

```
void saveThresholds() {  
    preferences.begin("settings", false);  
    preferences.putInt("soilMoisture", soilMoistureThreshold);  
    preferences.putInt("humidity", humidityThreshold);  
    preferences.end();  
}
```

Спочатку відкривається область пам'яті з назвою "settings" у режимі читання і запису за допомогою методу `preferences.begin()`. Далі нові порогові значення зберігаються у пам'ять за допомогою методів `putInt()`. Після цього область пам'яті закривається методом `preferences.end()`. Ця функція забезпечує збереження налаштувань навіть після перезавантаження плати, що дозволяє системі зберігати свої параметри між сеансами роботи.

3.3 Монтаж та перевірка працездатності системи

На даному етапі було проведено тестування системи спостереження параметрів навколишнього середовища на базі ESP32 C3, яка відображає дані через веб-інтерфейс. Це дозволяє користувачам отримувати поточні показники вологості ґрунту, вологості повітря та рівня води, а також налаштовувати порогові значення для автоматизації відповідних дій.

На рисунку 3.1, представлений веб-інтерфейс, розроблений для ESP32 C3, що показує дані датчиків у режимі реального часу. Інтерфейс відображає такі параметри, як вологість ґрунту, вологість повітря та рівень води, які є ключовими показниками для моніторингу стану навколишнього середовища. Значення цих параметрів оновлюються в реальному часі, що забезпечує точне та своєчасне інформування користувача про поточний стан системи.

ESP32 C3 Sensor Data

Soil Moisture: 300

Humidity: 75

Water Level: 411

Temperature: 21

Soil Moisture Threshold:

Humidity Threshold :

Рисунок 3.1 – Веб-інтерфейс системи контролю клімату та вологості ґрунту

На представленому рисунку вказано значення вологості ґрунту, яке становить 300, вологості повітря – 75, та рівня води – 411. Ці значення зчитуються з відповідних датчиків, підключених до плати ESP32 C3, і відображаються у веб-інтерфейсі. Інтерфейс також містить форми для введення порогових значень для вологості ґрунту та повітря [31]. Користувач може встановити ці значення, після чого система автоматично регулюватиме роботу водяного насоса та зволожувача відповідно до заданих параметрів.

Інформація про поточні значення датчиків відображається у вигляді тексту, що забезпечує простий та зрозумілий формат для користувача. Кожен параметр має свій рядок, що дозволяє легко зчитувати та аналізувати дані. Форма для введення порогових значень складається з двох полів введення, розташованих під відповідними мітками. Користувач може легко змінювати

значення порогів, вводячи нові дані у ці поля та натискаючи кнопку "Set Thresholds".

Важливим моментом тестування було переконатися, що система коректно реагує на зміни параметрів навколишнього середовища. Наприклад, якщо значення вологості ґрунту перевищує встановлений поріг, система автоматично вмикає водяний насос для поливу.

Тестування показало, що система стабільно працює при зміні умов та забезпечує точне зчитування даних з датчиків. Всі виміряні значення відображаються коректно і своєчасно, що підтверджує високу якість використовуваних компонентів та правильність їх налаштування. Веб-інтерфейс надає зручний спосіб взаємодії з системою, дозволяючи користувачам легко налаштовувати параметри та отримувати актуальну інформацію.

На рисунку 3.2 показано змонтовану систему контролю клімату та вологості ґрунту в домашній теплиці, яка включає різні компоненти. Основний елемент системи – плата ESP32 C3, яка розташована у захисному корпусі для забезпечення її безпеки та зменшення впливу зовнішніх факторів. До плати підключено декілька датчиків і виконавчих пристроїв.



Рисунок 3.2 – Змонтована система контролю

Давач вологості ґрунту знаходиться безпосередньо у ґрунті біля рослин. Він підключений до плати за допомогою проводів, що дозволяє зчитувати значення вологості ґрунту і передавати їх до системи для подальшої обробки. Також на зображенні присутній давач рівня води, який розташований у пластиковій ємності з водою. Цей давач дозволяє контролювати рівень води і забезпечує автоматичний полив рослин у разі потреби.

Важливою частиною системи є водяний насос, який може перекачувати воду з ємності до рослин. Насос керується платою ESP32 C3 на основі зчитаних даних з давачів. Якщо рівень вологості ґрунту знижується нижче встановленого порогу, система автоматично вмикає насос для забезпечення поливу. На фото видно трубки, через які вода подається до рослин.

Для підключення всіх компонентів використовуються електричні дроти та з'єднувачі. На рисунку видно, що дроти підключені до відповідних пінів плати та давачів, що забезпечує стабільну роботу системи і точність зчитування даних. Таке підключення дозволяє легко змінювати компоненти або додавати нові давачі у разі потреби.

На рисунку 3.3 зображено роботу системи поливу, яка активувалася через низький рівень вологості ґрунту. У даному випадку, значення вологості ґрунту знизилося нижче встановленого порогу 300, що викликало автоматичне включення водяного насоса.

На рисунку видно, як вода подається до рослини через трубку, що свідчить про активну роботу насоса.

Система моніторингу, виявила зниження вологості ґрунту за допомогою відповідного давача, який розташований у ґрунті біля рослин. Давач зчитує значення вологості і передає ці дані до плати, яка обробляє інформацію та приймає рішення про включення насоса. Це дозволяє підтримувати оптимальний рівень вологості для рослин, забезпечуючи їм необхідні умови для зростання.



Рисунок 3.3 – Робота системи поливу

Активний полив відбувається завдяки автоматичній системі, яка працює без участі користувача. Це значно полегшує процес догляду за рослинами, особливо в тих випадках, коли немає можливості постійно слідкувати за їх станом. Система реагує на зміни параметрів навколишнього середовища в режимі реального часу, забезпечуючи своєчасний полив та запобігаючи пересиханню ґрунту.

На даному етапі, окрім автоматичного поливу, система продемонструвала свою здатність ефективно контролювати рівень вологості повітря. На рисунку 3.4 видно, як система активувала зволоження повітря, через зниження рівня вологості нижче встановленого порогу 75%.



Рисунок 3.4 – Активація зволоження повітря

Це є критичним для забезпечення оптимальних умов для рослин, особливо в контрольованому середовищі.

Рослини розташовані в прозорій пластиковій ємності, яка виконує функцію міні-теплиці. Всередині ємності можна помітити невеликі краплі води, що свідчать про підвищення рівня вологості повітря. Це досягається завдяки активації системи зволоження, яка працює на основі даних, зчитаних з датчика вологості повітря. Датчик розташований у безпосередній близькості до рослин, що дозволяє точно вимірювати рівень вологості у міні-теплиці.

Система зволоження працює автоматично, реагуючи на зміни рівня вологості. Коли значення вологості повітря знижується нижче встановленого порогу 75%, система активує зволожувач, який починає розпилювати воду, підвищуючи вологість до необхідного рівня. Цей процес відбувається без участі

користувача, що значно спрощує догляд за рослинами і забезпечує стабільні умови для їх зростання.

Зволоження повітря є важливим моментом догляду за рослинами, особливо в умовах закритого простору, де рівень вологості може швидко змінюватися.

3.4 Висновок до третього розділу

Третій розділ кваліфікаційної роботи присвячений практичній реалізації та тестуванню системи контролю клімату та вологості ґрунту в домашній теплиці з використанням мікроконтролера ESP32 C3, який інтегровано із різними датчиками для моніторингу навколишнього середовища. Основна увага зосереджена на програмуванні ESP32 C3 у середовищі Arduino IDE, що використовує мову C/C++ для забезпечення гнучкості та високої продуктивності в контексті вбудованих систем.

Програмний код, який розроблений для системи, включає ініціалізацію апаратних ресурсів, постійне зчитування даних з датчиків та реагування на зміни згідно з заданими пороговими значеннями. Веб-інтерфейс, який реалізований через ESPAsyncWebServer, дозволяє користувачам налаштовувати порогові значення та відслідковувати реальний стан змінних умов навколишнього середовища.

Монтаж системи та її випробування показали, що обране обладнання та програмне забезпечення ефективно співпрацюють, забезпечуючи точність та надійність у моніторингу та контролі умов для досліджуваних параметрів. Зокрема, автоматизація регулювання вологості ґрунту та повітря демонструє можливість системи оперативно реагувати на зміни умов та підтримувати їх в оптимальних межах для потреб користувача.

РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Природне середовище і його забруднення

Природне середовище включає в себе всі живі і неживі компоненти, які взаємодіють між собою утворюючи унікальні екосистеми. Вода, повітря, ґрунт, рослини, тварини і мікроорганізми є основними елементами. Все починається з атмосфери, яка складається з різних газів таких, як азот, кисень і вуглекислий газ, ці гази необхідні для підтримки життя на Землі.

Атмосфера забезпечує захист від шкідливого ультрафіолетового випромінювання сонця а також підтримує кліматичні умови необхідні для життя. Вода є невід'ємною частиною природного середовища.

Вона присутня у океанах, морях, річках, озерах і підземних водах. Вода є життєво необхідною для всіх живих організмів. Вона використовується для пиття, приготування їжі, сільського господарства та промисловості.

Ґрунт також є важливою частиною природного середовища. Він забезпечує рослини поживними речовинами і підтримує їх ріст. Ґрунт також грає важливу роль у кругообігу води і вуглецю.

Рослини і тварини складають біотичну частину природного середовища. Вони взаємодіють між собою і з неживими компонентами утворюючи складні екосистеми. Рослини забезпечують їжу для тварин і людей, а також відіграють важливу роль у фотосинтезі виробляючи кисень.

Тварини у свою чергу відіграють важливу роль у підтримці екологічного балансу. Природне середовище піддається впливу різних факторів які можуть спричинити його забруднення.

Забруднення може бути природним або антропогенним тобто спричиненим людською діяльністю. Природні забруднення включають: вулканічні виверження, лісові пожежі і пилові бурі, які можуть викидати в атмосферу велику кількість газів і часток [20]. Проте антропогенні забруднення є найбільш значущими і тривалими. Джерела антропогенного забруднення

включають промисловість, транспорт, сільське господарство і побутову діяльність.

Промислові викиди є одним з основних джерел забруднення повітря і води. Вони включають викиди шкідливих газів таких як, діоксид сірки і оксиди азоту, а також важких металів і органічних речовин.

Транспорт також спричиняє значне забруднення викидаючи в атмосферу велику кількість вихлопних газів, які містять оксиди вуглецю, азоту і інші шкідливі речовини [20].

Сільське господарство спричиняє забруднення ґрунту і води через використання пестицидів і добрив, які можуть потрапляти у водні об'єкти і ґрунт.

Побутові викиди включають: сміття і відходи, які можуть забруднювати ґрунт і водні об'єкти. Наслідки забруднення природного середовища є різноманітними і можуть бути дуже серйозними [21]. Забруднення повітря може спричиняти проблеми зі здоров'ям у людей такі як респіраторні захворювання і серцево-судинні проблеми. Воно також може спричиняти кислотні дощі які пошкоджують рослини, ґрунт і водні об'єкти. Забруднення води може спричиняти загибель водних організмів і зниження якості питної води, що може мати серйозні наслідки для здоров'я людей. Забруднення ґрунту може призводити до зниження родючості ґрунту і зниження врожайності рослин. Боротися із забрудненням природного середовища можна різними способами [21].

Одним з найбільш ефективних способів є зменшення викидів шкідливих речовин через впровадження екологічно чистих технологій і використання альтернативних джерел енергії. Такими джерелами є сонячна вітрова і гідроенергія, які не спричиняють викидів шкідливих речовин. Важливим є також впровадження систем управління відходами, які дозволяють зменшити кількість сміття і забезпечити його правильну утилізацію або переробку. Освіта і підвищення обізнаності громадськості про проблеми забруднення природного середовища також є важливими заходами, які можуть сприяти зміні поведінки

людей і зменшенню забруднення. Іншими заходами є збереження природних ресурсів і захист природних екосистем через створення природоохоронних територій і заповідників ці території забезпечують захист різноманітних видів рослин і тварин і підтримують екологічний баланс [21]. Природне середовище є надзвичайно важливим для життя на Землі і його забруднення може мати серйозні наслідки для здоров'я людей і стану екосистем. Тому важливо вживати заходів для зменшення забруднення і збереження природного середовища для майбутніх поколінь.

4.2 Рекомендації щодо естетичного оформлення інтер'єру цеху, дільниці

Естетичне оформлення інтер'єру цеху чи дільниці є важливим аспектом, який впливає на продуктивність працівників, їхнє здоров'я та загальне враження від підприємства. Одним із перших кроків у створенні привабливого інтер'єру є вибір кольорової гами. Використання яскравих і теплих кольорів може стимулювати енергію і мотивацію працівників. Наприклад, жовтий колір сприяє підвищенню настрою та креативності, тоді як зелений заспокоює та сприяє концентрації. Важливо уникати надмірно яскравих кольорів, які можуть викликати втоми очей та знижувати продуктивність [22].

Наступним важливим аспектом є освітлення. Природне світло є найбільш сприятливим для робочого середовища, тому варто забезпечити максимальний доступ до нього. Це можна зробити за допомогою великих вікон або світлових люків. Якщо природне світло обмежене, варто використовувати якісне штучне освітлення з нейтральними відтінками, яке не викликати напруги очей [22]. Освітлення повинно бути рівномірним, без тіней, щоб забезпечити комфортні умови для роботи.

Організація простору також відіграє важливу роль в естетичному оформленні інтер'єру цеху чи дільниці. Простір повинен бути добре організованим та максимально зручним для працівників. Це включає правильне

розташування обладнання, робочих місць і зон відпочинку. Важливо уникати захаращення простору, що може призвести до стресу та зниження продуктивності [22].

Значну увагу слід приділити вибору меблів та обладнання. Вони повинні бути зручними, функціональними та відповідати ергономічним вимогам. Використання якісних матеріалів забезпечить довговічність меблів та обладнання, а також сприятиме створенню позитивного враження про підприємство [23]. Ергономічні стільці, столи і робочі станції допоможуть знизити ризик професійних захворювань і підвищити продуктивність працівників.

Декоративні елементи також можуть значно вплинути на загальний вигляд інтер'єру. Це можуть бути картини, плакати, зелень або інші елементи, які додають індивідуальності та стилю. Використання зелених рослин створює приємну атмосферу, покращує якість повітря і сприяє зниженню стресу. Варто також використовувати декоративні елементи, які відповідають тематиці підприємства або його продукції [23].

Звукова атмосфера є ще одним важливим аспектом естетичного оформлення. Надмірний шум може викликати стрес та знижувати продуктивність працівників [23]. Тому важливо забезпечити звуковий комфорт за допомогою звукоізоляційних матеріалів та обладнання, яке працює тихо. Використання музики або звуків природи може створити приємну атмосферу і покращити настрій працівників.

Температурний режим і вентиляція також важливі для створення комфортного робочого середовища. Оптимальна температура і свіже повітря сприяють підвищенню продуктивності та зниженню втоми. Варто забезпечити достатню кількість вентиляційних отворів або встановити кондиціонери для підтримання комфортної температури та забезпечення циркуляції повітря.

Оформлення зон відпочинку є важливою складовою естетичного оформлення інтер'єру. Вони повинні бути зручними і привабливими, щоб

працівники могли повноцінно відпочити під час перерви. Зручні меблі, зелень і приємна атмосфера сприяють відновленню сил і підвищенню мотивації.

Важливим аспектом є також чистота та порядок у цеху чи дільниці. Регулярне прибирання і підтримка чистоти сприяють створенню приємної атмосфери і знижують ризик нещасних випадків та професійних захворювань. Варто встановити системи управління відходами та забезпечити працівників засобами для підтримки чистоти на робочих місцях.

Нарешті, варто звернути увагу на безпеку та доступність інтер'єру. Всі робочі місця повинні бути безпечними і відповідати стандартам безпеки. Доступність до всіх зон повинна бути забезпечена для всіх працівників, включаючи тих, хто має обмежені фізичні можливості [23]. Це можна досягти за допомогою спеціальних пандусів, ліфтів та інших засобів.

Естетичне оформлення інтер'єру цеху чи дільниці є важливим елементом, який впливає на продуктивність працівників, їхнє здоров'я та загальний імідж підприємства. Правильний вибір кольорової гами, освітлення, організація простору, вибір меблів та обладнання, декоративні елементи, звукова атмосфера, температурний режим і вентиляція, оформлення зон відпочинку, підтримка чистоти і порядку, а також забезпечення безпеки та доступності допоможуть створити привабливий і комфортний робочий простір.

4.3 Висновок до четвертого розділу

У четвертому розділі кваліфікаційної роботи здійснено детальний аналіз впливу природного середовища на життєдіяльність людини та визначено основні напрямки охорони праці, що мають значення для забезпечення здорових та безпечних умов праці. Важливість збереження природного середовища і розвантаження працівників в умовах постійно зростаючого антропогенного навантаження на екосистему набуває особливої актуальності.

Основну увагу в розділі приділено оцінці стану природного середовища та його забруднення, що впливає на всі компоненти екосистеми та здоров'я

людей. Аналіз антропогенних та природних факторів, що спричиняють забруднення, вказує на необхідність розробки та впровадження ефективних технологій очищення та методів моніторингу якості елементів довкілля. Зокрема, важливим є використання альтернативних джерел енергії та розробка екологічно чистих технологій, що мінімізують вплив промисловості на природу.

Крім того, велике значення має психофізіологічне розвантаження працівників, що забезпечується через оптимізацію робочого простору, правильне використання кольорових рішень, освітлення, вентиляції та ергономічне оформлення робочих місць. Розвиток зон відпочинку, організація простору та естетичне оформлення робочого місця сприяють зниженню стресового навантаження, підвищують продуктивність та сприяють загальному покращенню здоров'я працівників..

ВИСНОВКИ

У даній кваліфікаційній роботі була розроблена система контролю клімату та вологості ґрунту для домашньої теплиці, що базується на використанні мікроконтролера ESP32 C3 та ряду датчиків для моніторингу та управління кліматичними умовами. В рамках роботи вивчались і аналізувались як промислові, так і аматорські системи, використання сучасних комунікаційних технологій та розробка програмного забезпечення, яке забезпечує взаємодію компонентів системи.

В першому розділі було здійснено аналіз сучасних систем моніторингу та управління, що дозволило ідентифікувати ключові технології та підходи, використані у промислових та аматорських системах. Основна увага була приділена комунікаційним можливостям та інтеграції систем з різними середовищами, що забезпечує гнучкість та адаптивність у відповідь на зміни умов.

У другому розділі детально розглядалися компоненти системи контролю, включаючи мікроконтролер ESP32 C3 і різні датчики для вимірювання параметрів середовища. Спроектвана система включала інтеграцію апаратних та програмних ресурсів, забезпечуючи точність та надійність у зборі даних і управлінні за допомогою розроблених електричних схем та використання веб-інтерфейсів для дистанційного контролю.

У третьому розділі була здійснена практична реалізація і тестування системи. Використання Arduino IDE та мови програмування C/C++ дозволило розробити програмне забезпечення, що ефективно управляє взаємодією між датчиками та виконавчими механізмами. Система продемонструвала свою ефективність у реальних умовах, забезпечуючи автоматизацію процесів поливу, регулювання вологості, що забезпечує оптимальні умови для вирощування рослин.

Проведені тестування роботи системи підтвердили, що система ефективно контролює кліматичні параметри та вологість ґрунту рослин. Що

довело її цінність для експлуатування в домашніх умовах. Система продемонструвала швидку здатність реагування на зміни параметрів та високу ефективність.

ПЕРЕЛІК ДЖЕРЕЛ

- 1 ControlByWeb. X-600M. URL: https://controlbyweb.com/wp-content/uploads/2024/01/X-600M_manual_v1.8.pdf (date of access: 21.06.2024).
- 2 Ларіоник, Р. В., & Луцик, Н. С. (2021). Комп'ютерна система для дистанційного контролю якості атмосферного повітря. Збірник тез доповідей X Міжнародної науково-практичної конференції молодих учених та студентів „Актуальні задачі сучасних технологій“, 1, 100-100.
- 3 Cloud-Based Remote Monitoring - Sentinels. Remote Monitoring Systems & Auto Dialers | Sensaphone. URL: <https://www.sensaphone.com/sentinels> (date of access: 21.06.2024).
- 4 Duda, O., Kunanets, N., Matsiuk, O., & Pasichnyk, V. (2022). Cloud-based IT Infrastructure for “Smart City” Projects. In Dependable IoT for Human and Industry (pp. 389-409). River Publishers.
- 5 Bodnarchuk, I., Duda, O., Kharchenko, A., Kunanets, N., Matsiuk, O., & Pasichnyk, V. (2020). Choice Method of Analytical Platform for Smart City (No. 4374). EasyChair.
- 6 Argus LIVE. Argus. URL: <https://arguscontrols.com/products-and-solutions/control-systems/live-software> (date of access: 21.06.2024).
- 7 Дуда, О., & Станько, А. (2023). ОРГАНІЗАЦІЯ ПРОЦЕСІВ СПОСТЕРЕЖЕННЯ ДАНИХ ЗАСОБАМИ ІОТ-ПРИСТРОЇВ У «РОЗУМНИХ МІСТАХ». Матеріали конференцій МЦНД, (06.10. 2023; Дніпро, Україна), 76-78.
- 8 Дуда, О. М. (2020). Інформаційні технології супроводу процесів в міських ресурсних та соціокомунікаційних мережах (Doctoral dissertation, Тернопільський національний технічний університет ім. Івана Пулюя).
- 9 Elaydi, H. (2017). An automated irrigation system for greenhouses. American Journal of Electrical and Electronic Engineering, 5(2), 48-57.

10 Turk, A. M., Gunal, E. S., & Gurel, U. (2016). An automation system design for greenhouses by using DIY platforms. In *The international conference on science, ecology and technology (Iconsete'2015–Vienna)* (pp. 257-266).

11 Васи́лишин, В. В., Тимо́щук, В. Д., Кітчак, Н. Ю., & Луцик, Н. С. (2023). Аналіз характеристик та застосування мікроконтролерів Attiny85, Atmega8, RP2040. *Матеріали XII Міжнародної науково-практичної конференції молодих учених та студентів „Актуальні задачі сучасних технологій“*, 420-420.

12 Kochan, V., Kunanets, N., Pasichnyk, V., Roshchupkin, O., Sachenko, A., Turchenko, I., ... & Matsiuk, O. (2019, September). Sensing in IoT for Smart City Systems. In *2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)* (Vol. 2, pp. 579-585). IEEE.

13 Тимо́щук, В., Васи́лишин, В., Мудрий, І., & Луцик, Н. (2023). Огляд та порівняння протоколів передачі інформації в IoT. *Матеріали XI науково-технічної конференції „Інформаційні моделі, системи та технології“*, 188-188.

14 Shirsath, D. O., Kamble, P., Mane, R., Kolap, A., & More, R. S. (2017). IoT based smart greenhouse automation using Arduino. *International Journal of Innovative Research in Computer Science & Technology*, 5(2), 234-238.

15 Тимо́щук, В., & Тимо́щук, Д. (2023). ПОРІВННЯ КАНАЛІВ ПЕРЕДАЧІ ІНФОРМАЦІЇ. *Матеріали конференцій МНЛ*, (23 червня 2023 р., м. Дніпро), 112-114.

16 Kryazhych, O., Itskovych, V., Iushchenko, K., Hrytsyshyna, V., Bruvier, D., Nykytyuk, V., & Bodnarchuk, I. (2023). The use of abstract Moore automaton to control the sensors of a service-oriented alarm and emergency notification network. *Вісник Тернопільського національного технічного університету*, 109(1), 111-120.

17 MQTT Core Concepts | EMQX Docs. EMQX Documentation. URL: <https://docs.emqx.com/en/emqx/latest/messaging/mqtt-concepts.html> (date of access: 21.06.2024).

18 Тимошук, В., Карташов, В., Королюк, Р. І., & Рубен, Т. (2022). Огляд протоколів керування для побудови автоматизованих систем віддаленого управління. Матеріали X науково-технічної конференції „Інформаційні моделі, системи та технології “Тернопільського національного технічного університету імені Івана Пулюя, 143-144.

19 The WebSocket API (WebSockets) - Web APIs | MDN. MDN Web Docs. URL: https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API (date of access: 21.06.2024).

20 Петрик, О. М., Суховерша, В. О., & Марценко, С. В. (2022). Дослідження ролі IoT-технологій в промислових комп'ютерних мережах. Матеріали XI Міжнародної науково-практичної конференції молодих учених та студентів „Актуальні задачі сучасних технологій “, 138-138.

21 1 Zigbeealliance. Zigbee. URL: <https://zigbeealliance.org/wp-content/uploads/2019/11/docs-05-3474-21-0csg-zigbee-specification.pdf> (date of access: 21.06.2024).

22 Get Started - ESP32-C3 - – ESP-IDF Programming Guide v5.2.2 documentation. Technical Documents | Espressif Systems. URL: <https://docs.espressif.com/projects/esp-idf/en/stable/esp32c3/get-started/index.html> (date of access: 21.06.2024).

23 Duda, O., Martsenko, S., Matsiuk, O., Kunanets, N., & Pasichnyk, V. (2020). Building secure Urban information systems based on IoT technologies. In CEUR Workshop Proceedings (pp. 317-328).

24 DHT11–Temperature and humidity sensor. Components101. URL: <https://components101.com/sensors/dht11-temperature-sensor> (date of access: 21.06.2024).

25 FC-28 Soil Moisture Sensor Module - Datasheet Hub. Datasheet Hub. URL: <https://www.datasheethub.com/fc-28-soil-moisture-sensor-module/> (date of access: 21.06.2024).

26 Water Level Sensor – Biomaker.org. Biomaker.org. URL: <https://www.biomaker.org/block-catalogue/2021/12/17/water-level-sensor-tzt-water-level-sensor> (date of access: 21.06.2024).

27 RS360SH. Datasheets. URL: <http://www.microelectronicos.com/datasheets/RS360SH.pdf> (date of access: 21.06.2024).

28 IRF520 Datasheet. Datasheets PDF - Integrated circuits, Transistor, Diodes, Triacs, and other Semiconductors Datasheet. URL: <https://datasheetspdf.com/datasheet-pdf/283678/IRF520.html> (date of access: 21.06.2024).

29 Sonoff IP66 - Waterproof Case | SONOFF Official. SONOFF Official. URL: <https://sonoff.tech/product/accessories/ip66/> (date of access: 21.06.2024).

30 Arduino ide documentation. arduino.cc. URL: <https://docs.arduino.cc/software/ide/> (date of access: 21.06.2024).

31 Ларіоник, Р., Луцик, Н., & Паламар, А. М. (2021). Система для моніторингу якості атмосферного повітря на базі IoT. Матеріали ІХ науково-технічної конференції „Інформаційні моделі, системи та технології“, 119-119.

32 Вбивча природа: як забруднення навколишнього середовища впливає на здоров'я. Новини України - останні новини України сьогодні - УНІАН. URL: <https://www.unian.ua/ecology/1304769-vbivcha-priroda-yak-zabrudnennya-navkolishnogo-seredovischa-vplivae-na-zdorovya.html> (date of access: 21.06.2024).

33 Сайт журналу «Геоінформатика» Головна - Сайт журналу «Геоінформатика». URL: https://www.geology.com.ua/wp-content/uploads/2015/05/15_Stakhiv.pdf (date of access: 21.06.2024).

34 7 секретів від дизайнера інтер'єру робочих просторів | Продизайн. Продизайн. URL: <https://prodesign.in.ua/2023/08/7-sekretiv-vid-dyzajnera-interyeru-robochuh-prostoriv/> (date of access: 21.06.2024).

35 5 підказок щодо оформлення інтер'єру робочого місця . arthuss. URL: <https://www.arthuss.com.ua/books-blog/5-pidkazok-shchodo-oformlennya-interyeru-robochoho-mistsya> (date of access: 21.06.2024).

ДОДАТКИ

Код для ESP32 C3

```
#include <Adafruit_GFX.h>
#include <Adafruit_ST7735.h>
#include <SPI.h>
#include <DHT.h>
#include <UIPEthernet.h>
#include <PubSubClient.h>
#include <EmonLib.h> #include <WiFi.h>
#include <ESPAsyncWebServer.h>
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <Preferences.h>

#define SOIL_MOISTURE_PIN 1
#define WATER_PUMP_PIN 2
#define HUMIDITY_SENSOR_PIN 3
#define FAN_PIN 4
#define WATER_LEVEL_PIN 5

int soilMoistureThreshold = 300;
int humidityThreshold = 75;

DHT dht(HUMIDITY_SENSOR_PIN, DHT11);

int soilMoistureValue = 0;
float humidity = 0.0;
int waterLevelValue = 0;

const char* ssid = "SSID";
const char* password = "XXXXXX";

AsyncWebServer server(80);
```



```
Preferences preferences;

void setup() {
  Serial.begin(115200);

  pinMode(SOIL_MOISTURE_PIN, INPUT);
  pinMode(WATER_PUMP_PIN, OUTPUT);
  pinMode(FAN_PIN, OUTPUT);
  pinMode(WATER_LEVEL_PIN, INPUT);

  dht.begin();

  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
  Serial.println("Connected to WiFi");
  Serial.print("IP Address: ");
  Serial.println(WiFi.localIP());

  preferences.begin("settings", false);
  soilMoistureThreshold = preferences.getInt("soilMoisture",
soilMoistureThreshold);
  humidityThreshold = preferences.getInt("humidity",
humidityThreshold);
  preferences.end();

  server.on("/", HTTP_GET, [] (AsyncWebServerRequest *request) {
    String html = "<html><head><style>";
    html += "body { font-family: Arial, sans-serif; background-
color: #e6ffe6; }";
```

```

    html += ".container { max-width: 600px; margin: auto; padding:
20px; background-color: #ffffff; box-shadow: 0 0 10px rgba(0, 0,
0, 0.1); }";
    html += "h1 { color: #4CAF50; }";
    html += "p { font-size: 18px; }";
    html += "form { margin-top: 20px; }";
    html += "label { display: block; margin-bottom: 10px; }";
    html += "input[type='number'] { width: 100%; padding: 8px;
margin-bottom: 20px; border: 1px solid #ccc; border-radius: 4px;
}";
    html += "input[type='submit'] { background-color: #4CAF50;
color: white; padding: 10px 20px; border: none; border-radius:
4px; cursor: pointer; }";
    html += "input[type='submit']:hover { background-color:
#45a049; }";
    html += "</style></head><body>";
    html += "<div class='container'>";
    html += "<h1>ESP32 Sensor Data</h1>";
    html += "<p>Soil Moisture: " + String(soilMoistureValue) +
"</p>";
    html += "<p>Humidity: " + String(humidity) + "</p>";
    html += "<p>Water Level: " + String(waterLevelValue) + "</p>";
    html += "<form action='/set' method='POST'>";
    html += "    <label for='soilMoisture'>Soil Moisture
Threshold:</label>";
    html += "    <input type='number' id='soilMoisture'
name='soilMoisture' value='" + String(soilMoistureThreshold) +
"'>";
    html += "    <label for='humidity'>Humidity Threshold:</label>";
    html += "    <input type='number' id='humidity' name='humidity'
value='" + String(humidityThreshold) + "'>";
    html += "    <input type='submit' value='Set Thresholds'>";
    html += "</form>";
    html += "</div></body></html>";
    request->send(200, "text/html", html);

```

```

});

server.on("/set", HTTP_POST, [] (AsyncWebServerRequest *request){
    if (request->hasParam("soilMoisture", true) && request-
>hasParam("humidity", true)) {
        soilMoistureThreshold = request->getParam("soilMoisture",
true)->value().toInt();
        humidityThreshold = request->getParam("humidity", true)-
>value().toInt();

        preferences.begin("settings", false);
        preferences.putInt("soilMoisture", soilMoistureThreshold);
        preferences.putInt("humidity", humidityThreshold);
        preferences.end();

        request->send(200, "text/html", "<html><body><h1>Settings
Updated</h1><p><a href='/'>Go back</a></p></body></html>");
    } else {
        request->send(400, "text/html", "<html><body><h1>Invalid
Input</h1></body></html>");
    }
});

server.begin();
}

void loop() {
    soilMoistureValue = analogRead(SOIL_MOISTURE_PIN);
    Serial.print("Soil Moisture: ");
    Serial.println(soilMoistureValue);

    humidity = dht.readHumidity();
    Serial.print("Humidity: ");
    Serial.println(humidity);
}

```

```
waterLevelValue = analogRead(WATER_LEVEL_PIN);  
Serial.print("Water Level: ");  
Serial.println(waterLevelValue);  
  
if (soilMoistureValue > soilMoistureThreshold) {  
    digitalWrite(WATER_PUMP_PIN, HIGH);  
    delay(3000);  
    digitalWrite(WATER_PUMP_PIN, LOW);  
}  
  
if (humidity > humidityThreshold) {  
    digitalWrite(FAN_PIN, HIGH);  
} else {  
    digitalWrite(FAN_PIN, LOW);  
}  
  
delay(2000);  
}
```